

AD-A171 306

THE INFLUENCE OF LARGE-SCALE COMPUTING ON AIRCRAFT  
STRUCTURAL DESIGN(U) ADVISORY GROUP FOR AEROSPACE  
RESEARCH AND DEVELOPMENT NEUILLY-SUR-SEINE (FRANCE)  
APR 86 AGARD-R-726

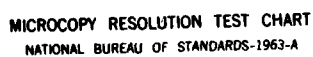
1/1

UNCLASSIFIED

F/G 9/2

NL

END  
DATE  
FILMED  
10-86



AGARD

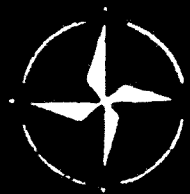
ADVISORY GROUP FOR AEROSPACE RESEARCH & DEVELOPMENT

THE AGARD GROUP IS A JOINT VENTURE OF THE NATO MEMBER STATES

AGARD REPORT No. 726

# The Influence of Large-Scale Computing on Aircraft Structural Design

NORTH ATLANTIC TREATY ORGANIZATION



NORTH ATLANTIC TREATY ORGANIZATION  
ADVISORY GROUP FOR AEROSPACE RESEARCH AND DEVELOPMENT  
(ORGANISATION DU TRAITE DE L'ATLANTIQUE NORD)

AGARD Report No. 726  
**THE INFLUENCE OF LARGE-SCALE COMPUTING  
ON AIRCRAFT STRUCTURAL DESIGN**

Accession For	
NTIS CLASS	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/ _____	
Availability Codes	
Dist	Avail and/or Special
<b>A-1</b>	



## THE MISSION OF AGARD

The mission of AGARD is to bring together the leading personalities of the NATO nations in the fields of science and technology relating to aerospace for the following purposes:

- Exchanging of scientific and technical information;
- Continuously stimulating advances in the aerospace sciences relevant to strengthening the common defence posture;
- Improving the co-operation among member nations in aerospace research and development;
- Providing scientific and technical advice and assistance to the Military Committee in the field of aerospace research and development (with particular regard to its military application);
- Rendering scientific and technical assistance, as requested, to other NATO bodies and to member nations in connection with research and development problems in the aerospace field;
- Providing assistance to member nations for the purpose of increasing their scientific and technical potential;
- Recommending effective ways for the member nations to use their research and development capabilities for the common benefit of the NATO community.

The highest authority within AGARD is the National Delegates Board consisting of officially appointed senior representatives from each member nation. The mission of AGARD is carried out through the Panels which are composed of experts appointed by the National Delegates, the Consultant and Exchange Programme and the Aerospace Applications Studies Programme. The results of AGARD work are reported to the member nations and the NATO Authorities through the AGARD series of publications of which this is one.

Participation in AGARD activities is by invitation only and is normally limited to citizens of the NATO nations.

The content of this publication has been reproduced  
directly from material supplied by AGARD or the authors.

Published April 1986

Copyright © AGARD 1986  
All Rights Reserved

ISBN 92-835-1522-6



Printed by Specialised Printing Services Limited  
40 Chigwell Lane, Loughton, Essex IG10 3TZ

CONTENTS

	Reference
<b>SUMMARY</b> by A.J.Morris	S
➤ <b>MODERN TRENDS IN AIRCRAFT STRUCTURAL DESIGN ;</b> by H.Gödel, H.Hörnlein, D.Keppeler and O.Sensburg	1
➤ <b>THE EFFECTIVE USE OF COMPUTING POWER IN STRUCTURAL ANALYSIS ;</b> by I.C.Taig	2
<b>COMPUTER TECHNOLOGY AND ENGINEERING – PAST, PRESENT, AND 1995 ;</b> by N.A.Radovcich	3
<b>THE NEEDS OF THE USAF AIR LOGISTICS CENTERS FOR LARGE SCALE STRUCTURAL ANALYSES AND SUPPORTING DATA</b> by T.F.Christian, Jr	4
<b>IMPACT OF COMPUTER ADVANCES ON FUTURE FINITE ELEMENT COMPUTATIONS</b> by R.E.Fulton	5

## LARGE SCALE COMPUTING TRENDS AND POTENTIAL AEROSPACE APPLICATIONS

### SUMMARY

Professor A.J. MORRIS  
College of Aeronautics  
Cranfield Institute of Technology  
Cranfield, Bedford  
United Kingdom

#### 1. Introduction

*→ This report reviews*

The papers contained herein represent an attempt to review the impact of the modern revolution in computing hardware in aerospace design, and were presented to the AGARD/SMP activity on Large Scale Computing in Aircraft Design in San Antonio, Texas, USA, April 1985. They identified several areas of potential application relevant to aeronautical design ranging from applications of artificial intelligence to the use of computing power for alleviating management control problems. The studies indicate that the radical application of large scale computing can have a major impact on the future use of such varied aspects as automated design techniques, the development of extremely user friendly programs, the preservation of knowledge and advanced monitoring procedures. In this short introductory paper an effort is made to synthesise these contributions in a concise manner leaving later chapters to amplify the themes outlined below.

#### 2. Trends in Computing Power

The power and cost effectiveness of modern computers is increasing due to a variety of innovative technological advances which have taken place in recent years and which will be developed in the near future. The impact of VLSI on computer memory has been dramatic to the extent that very large amounts of memory are now able to be lodged on a single chip. Currently we are looking at the eight megabyte memory chip and developments in this area will undoubtedly lead to even larger single chip memory boards. This technology clearly advances the speed of the computer in allowing rapid deposition of data on to memory and subsequent rapid accession of this data into the main computing stream. Nesting this type of memory board into pipelining or DAP computers is inevitably going to increase the computing power beyond those currently being exhibited by Cray II and comparable super computers. On the smaller scale the power of micro computers and micro processors is similarly increasing at a very rapid pace. The full 32 byte microprocessor with a cycle frequency of 20 MHz is now available and in the future we would look to this type of machine being able to perform 2 megaflops of double precision and, in pipelining configuration, up to 15 megaflops. By using this type of microprocessor power it is becoming possible to define a floating point computing engine with a given power which can interface with an existing or new main frame computer. It has therefore become possible to define a computer processing system which is tailored to a specific requirement in aerospace or other fields. Indeed, it is possible to argue that with the reduction in cost it may well be appropriate to define the system which is appropriate not just to a particular domain of application such as aerospace engineering but to a precise application within that domain, such as Artificial Intelligence, or Advanced Automated Design. We may therefore look to a multi-operating system in which a host computer interfaces with a range of alternative computing systems appropriately configured for the basic programming need. The host computer can be thought of either as a serial type computer of the type commonly used in day to day applications at the present time or it could also be an advanced pipelining machine in its own right which is able to more intimately dovetail its form of operation with the parasite computers.

This type of approach would require that the computer hardware manufacturers were prepared to produce computers suitable to this type of aerospace need. Because the aerospace industry is a very small sector of the market it was made clear that the industry could have to be precise in defining the problem areas where developments were required; a coordinated approach would be essential in order to get an adequate response from the hardware manufacturers.

AGARD can be viewed as one forum in which this type of coordinated approach could be developed and through it pressure brought to bear on the appropriate hardware developers. The remaining sections in this short paper endeavour to define some of these areas which were highlighted at the San Antonio Meeting and which are reported on in the following articles.

#### 3. Integrated Design

There is a growing requirement which is now well recognised for the design process in the future to integrate many of the disciplines which are used in creating an aerospace vehicle. The various structural and aeronautical aspects will have to be considered jointly if the interaction of the one upon the other is to be fully exploited, such as when new materials, eg carbon fibre reinforced plastics are used extensively in the design process. Other drivers in this area come from the need to create large flexible structures which often have to be built to very high tolerances for space use.

This gives rise to the complex CAD/CAM requirement in which the lessons learned from IPAD and other similar projects would have to be incorporated. The integrated design software would accommodate a multiple view of the data in which, for example, the design would sometimes be regarded from a purely structural standpoint and at other times would take full account of aeronautical fatigue and management requirements. This process would necessarily require a multi-level data structure to represent the total design. Output from such a relational database system would then be linked to several discipline orientated programs taking into account structural optimisation methods, aerodynamic loads, finite element stress calculations etc. Modern database systems can handle this kind of design program and it is quite feasible to think in terms of creating a suitable finite element machine to do the necessary non-linear dynamic or other analysis calculations. This view of the design process is expanded by the following paper and is supported by the MBB view that improved quality and production do require integrated software going from mission analysis through aerodynamic calculations, geometry considerations and even incorporating wind tunnel analysis. The MBB approach sees the whole of this procedure being wrapped around optimisation algorithms which are able to take account of active control concepts, look at gust alleviation procedures, take into account flutter calculations and so forth.

The generation, development, and exploitation of this type of large scale computing program requires massive computational power. The modern large scale computers of the Cray II type would be able to handle certain of the issues by this type of approach but clearly would not be able to cover the full range of computational and data management required to be able to combine most of the design aspects thought necessary in such an integrated program. Several contributors see integrated programs as being one of the drivers and motivators to the exploitation and future generation of the large scale computing capacity touched on in section 2.

#### 4. Management and Control

One of the major problems in controlling and organising a large fleet of aircraft is to ensure that the structural integrity of each individual aircraft is maintained at all times throughout its useful life. Currently this problem is being faced by the US Air Force and the Aircraft Structural Integrity Program and Damage Tolerance Analysis Procedures have been set up to endeavour to follow each aircraft on a flight by flight analysis. This is supported by the Aircraft Information Retrieval System which is an advanced management tool allowing each aircraft to be tracked and a repair program to be set up to effectively maintain the integrity of aircraft in service at all times. It is also intended to allow repair schemes to be created and clearances given at a moment's notice for aircraft that might be required for combat usage. The main problem in generating this type of complex software to manage the flight programs of individual aircraft lies in creating appropriate data bases using pre and post processors. The concept behind this type of program is fairly clear and straightforward. However the problem of inputting the required data in a suitably structured form and subsequently retrieving it in an effective manner is far from solved and will require very large computing capabilities.

Whilst it is possible to believe that the current large scale computers are able to handle the data required for the ASIP and the AIRS programs, when consideration is given to the extension of this type of philosophy to systems involving not only structural aspects but also avionics systems, engines, etc. it is clear that the current range of computers is inadequate for the task to be handled.

#### 5. Artificial Intelligence

The use of integrated design programs appears to offer the user the easy method for forming elaborate and complex design and analyses. However modern CAD/CAM systems are extremely elaborate and complex themselves and require very specialist knowledge to be effectively used. Extending this type of complexity to the integrated design program would require such elaborate specialisation on the part of the user that the programs in many ways would be self defeating since they would now be inaccessible to the design community. In order to make very advanced computer programs accessible to the design engineer attention is now focussing on the use of artificial intelligence as a way of holding knowledge within the computer and instructing and guiding the user of complex programs in their use and giving assistance to avoid abuse. The approach proposed by British Aerospace extends this concept to taking into account the fact that many aircraft design engineers are now reaching retirement and their knowledge will be lost to the companies unless action is taken. The idea proposed by BAe is therefore to create special programs which will preserve the company's knowledge and also allow advanced programs to be used within the same network of artificially intelligent programs.

The use of complex languages such as PROLOG and LISP places a great strain on computers and in many cases dictates that a high level of parallel computing should be performed. In order to create effective programs in the aircraft design areas extremely versatile computers will be required. The concept of the 'conformable' computer is one which is advanced in this area. This concept envisages a computer with a multiple processing capacity which is itself able to make decisions about when to run the programs sequentially and when to co-routine. Such a computer would also be able to support the heavy numerical computation requirements laid down by the integrated design programs, calling up structural optimisation and finite element programs as well as being able to support the AI aspects which would be lodged within this type of program in future.



#### 6. Conclusion

The papers presented in the sequel clearly emphasise that large scale computing hardware has a major role to play in future aeronautical design. It is also clear that certain advances will hinge not only on the availability of current advanced computers but on further advances that are foreseen in the near to medium term. An important aspect which emerges from the papers is that the old idea of regarding aircraft design within clearly defined disciplines such as structures, aerodynamics, systems, etc, will not be applicable when large scale computing hardware is freely available and exploited for the design of advanced aerospace vehicles.

## MODERN TRENDS IN AIRCRAFT STRUCTURAL DESIGN

by

H. Gödel  
H. Hörnlein  
D. Keppeler  
O. Sensburg

MESSERSCHMITT-BÜLKOW-BLOHM GmbH.  
Helicopter and Airplane Division  
P.O. Box 801160, 8 Munich 80  
W.-Germany

Summary

This paper deals mainly with the usage of supercomputers in structural optimization. To employ an integrated design strategy where several disciplines of aircraft structural design must cooperate - mainly loads, flight control system, stress and static aeroelastics- optimization program architectures become more and more complex. It is also necessary to represent and optimize the structure - especially in highly loaded areas - with a large number of finite elements in order to achieve weight reductions. It is shown that supercomputers could make the automated structural design problem feasible.

1. INTRODUCTION

In the first applications of structural optimization it was tried to put the intuitive physical criteria of optimality into algorithms. This procedure gave nonlinear relations which had to be truncated in order to get easy to use redesign formulas. This approach is the well-known Optimality Criteria method (OC). The OC methods are appropriate to provide the "optimal design" of very special structural problems under severe restricting assumptions. The stress rationing method, for example leads to a fully stressed design (FSD), but this is the optimal design only, if the structure is statically determinate, if the used material has the same properties for all elements and if only stress constraints are imposed. Not even the deflection constraints which are in linear relation to the stress constraints can be handled simultaneously by this approach.

The development of further redesign formulas in order to deal with distinct constraints leads to an interactive procedure. This approach however is questionable because it is a priori not known which constraints are active in the optimal design.

The simultaneous consideration of all constraints was not possible until the Mathematical Programming methods (MP) were employed. The MP methods which are independent of physics, supply the necessary generalization for the treatment of physically different constraints. The development of the MP methods however did not aim at the solution of nonlinear physical problems but was applied only on linear, quadratic or at best on nonlinear convex models. For that reason the application of MP methods is also restricted to some assumptions. No global theory for multimodal problems exists so only local solutions can be found. It is necessary for the future development of structural optimization to deal with the specific properties of "real life" problems and to revise the MP methods.

2. STRUCTURAL OPTIMIZATION USING MATHEMATICAL PROGRAMMING

The task of structural optimization consist in the determination of optimal values of design variables  $x \in \mathbb{R}^n$  which minimize a selected objective function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ , while satisfying imposed constraints  $g_j(x) \geq 0$ ;  $j \in J$ . A typical example is minimizing weight subject to stress constraints. The problems may be classified by the type of chosen design space. Three different types of design variables are usually considered, see Fig. 1. Each class of design space requires different appropriate solver. Ideally all kinds of design variables should be handled simultaneously. In practice, this is done consecutively because of the differences in numerical behaviour.

Another kind of classification arises from the different physical properties of the constraints, but these problems must be handled simultaneously, see Fig. 1.

Classification by the type of chosen design space

- . sizing variables  $\Rightarrow$  actual design of members
- . geometrical variables  $\Rightarrow$  shape design of structure
- . topological variables  $\Rightarrow$  arrangements of members

Classification by the kind of considered constraints

- . static  $\Rightarrow$  stress, displacement, buckling
- . dynamic  $\Rightarrow$  frequency, response
- . aeroelastic  $\Rightarrow$  efficiency, flutter-, divergence speed

FIG. 1 CLASSIFICATION OF PROBLEMS

The general nonlinear programming (NLP) problem

NLP
$\begin{aligned} \min f(x) \quad & x \in \mathbb{R}^m \\ \text{s.t. } g_j(x) \geq 0 \end{aligned}$

is usually attempted to be solved by two different iterative procedures.

MP	Mathematical Programming
$x^{v+1} := x^v + \alpha^v s^v \quad v \in \mathbb{N}$	
where $s^v$ is search direction and $\alpha^v$ is step size	

OC	Optimality Criteria
$x_i^{v+1} := \psi_i(x^v) \quad v \in \mathbb{N}$	
$i = 1, \dots, m$	
where $\psi$ is an appropriate recurrence relation	

FIG. 2 TWO SCHOOLS OF THOUGHT : MP-OC

In general, the structural optimization problem is well modelled as a nonlinear programming problem (NLP), see Fig. 2. As mentioned in the introduction there are two schools of thought and there is an age-old dispute between the supporters of Mathematical Programming (MP) and Optimality Criteria (OC). In reality there is no difference in principle between these schools of thought. Duality makes it simply clear that the OC school has always tried to solve the dual problem. The recurrence relations (redesign formulas) used in the OC methods offer nothing else than the iterative fulfilment of the necessary first order conditions (Kuhn-Tucker Conditions) for local optimality. However, these non-linear conditions often were, and still are, greatly simplified in order to obtain recurrence relations that can easily be dealt with. It is, therefore, no real surprise that the solutions found represent mainly mere approximations of the true optimum.

The observation that the OC methods also belong to the MP methods is not new. C.Fleury and others repeatedly emphasized this years ago and appealed for the settling of the dispute between these only seemingly different schools of thought.

The MP methods are usually divided into two categories, see Fig. 3. The use of Transformation Methods is as popular now as it was before. In this case, however, the idea of "augmented Lagrangian" - also known as Method of Multipliers - seems to find acceptance. The arising unconstrained problems are solved using the well known quasi Newton methods (DFP, BFGS). The self scaling variable metric algorithms, which were put forward by S.S. Oren and D.G. Luenberger could still produce improvements. This scaling possibly makes the quasi Newton method more stable against non-exact line search, since the deterioration of the condition of the updated inverse Hessian is prevented.

#### Frequently used MP methods

All MP-methods are based on solving the local Kuhn-Tucker conditions. These methods are usually divided into two categories:

#### Transformation methods

- . Barrier functions
- . Penalty functions
- . Methods of multipliers (augmented Lagrangian)

#### Primal methods

- . Sequential linear programming (SLP)
- . Recursive quadric programming (RQP)
- . Gradient projection methods
- . Generalized reduced gradients (GRG)
- . Method of feasible directions (MFD)

FIG. 3 MATHEMATICAL PROGRAMMING METHODS

Without going into further details, let it be mentioned that, as the Primal Methods, besides the Sequential Linear and Recursive (sequential) Quadratic Programming (SLP, RQP), the methods of Generalized Reduced Gradients (GRG) as well as those of Projected Gradients, are also being successfully used. Zoutendijk's idea (Method of Feasible Directions) is still very often applied.

Designing a structural optimization program system there are some items in view of mathematical programming which should be regarded.

- o Development of in-house programs gives familiarity with the details.
- o Sensitivity analysis plays a crucial part in structural optimization (CPU time consuming).
- o Variable linking is necessary to handle large scale, symmetric and/or antisymmetric problems with a lot of fixed variables and production requirements respectively.
- o Temporary disregard of "inactive" constraints by suitable "active set" strategies saves computer time.
- o Use explicit constraints approximations for active constraints.
- o Scaling of constraints and variables makes the numerical problems well behaved.

### 3. STRUCTURAL OPTIMIZATION IN THE GENERAL DATA FLOW

The use of structural optimization tools during the preliminary design stage of an advanced aircraft gives the following potential improvements.

- o satisfying the requirements of recent aircrafts.
- o minimizing the objective (weight)
- o increasing the quality of products.
- o shortening the development phase.
- o increasing chances of the company in competition

In order to do this, an appropriate mathematical programming procedure has to be embedded in the general data flow, which is depicted in Fig. 4 and Fig. 5.

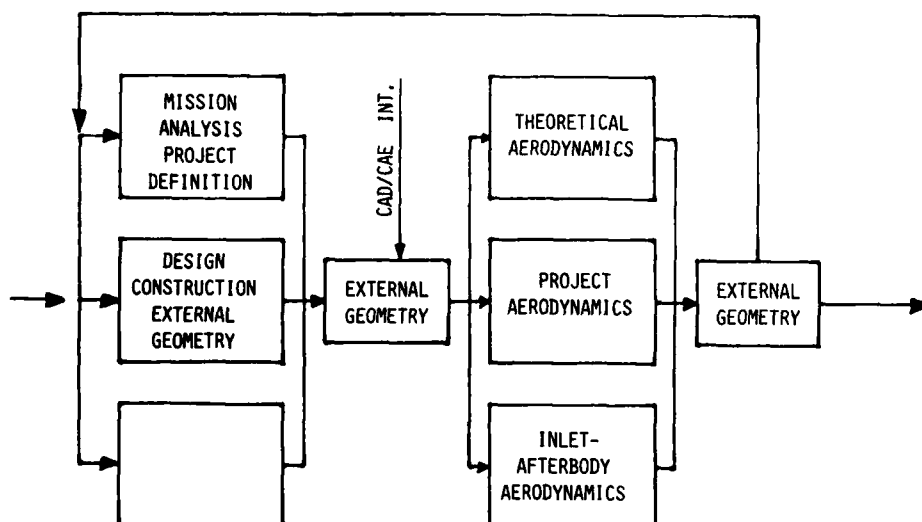


FIG. 4 EXTERNAL GEOMETRY IN DATA FLOW

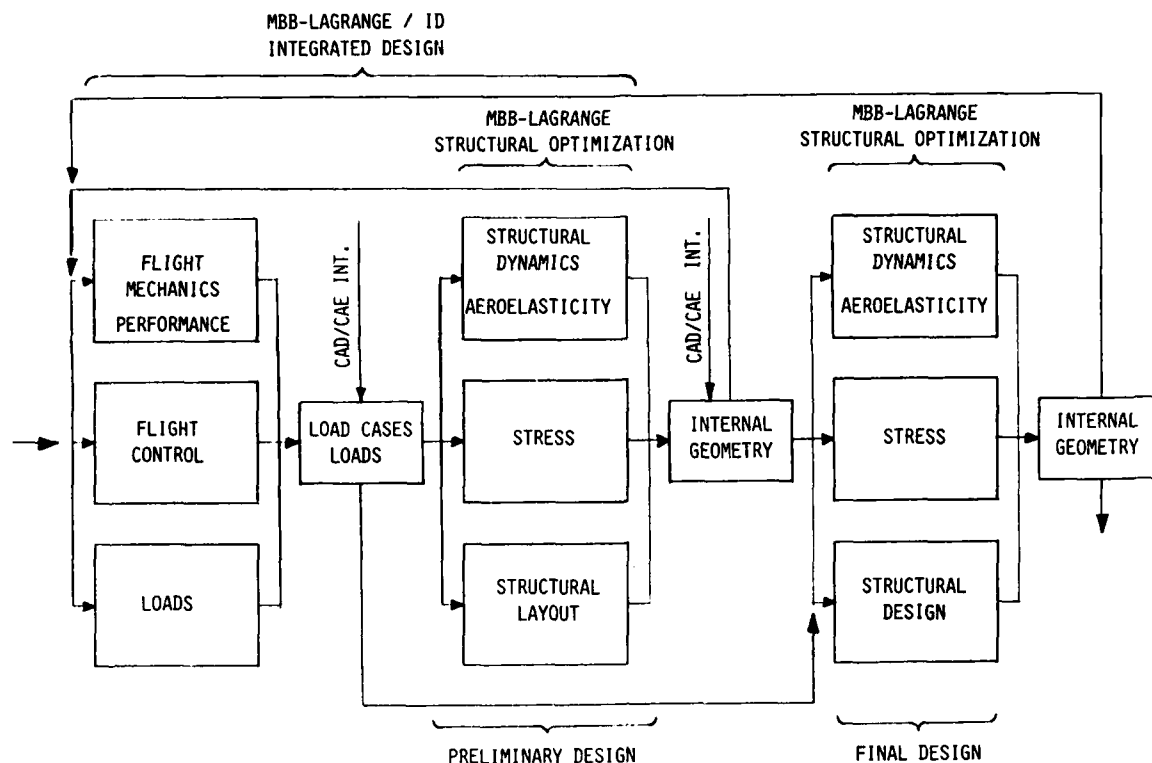


FIG. 5 GENERAL DATA FLOW

These figures show a typical flow of geometric, aerodynamic, structural and other data which are used during the design phase of an aircraft. The improved productivity is a result of the integrating effects of the structural optimization. Shorter time of development would be expected and fewer data transfer would go wrong.

At the present time the development of a recent aircraft is influenced by new techniques, such as flutter suppression, CCV-configuration, gust load alleviation etc, see Fig. 6.

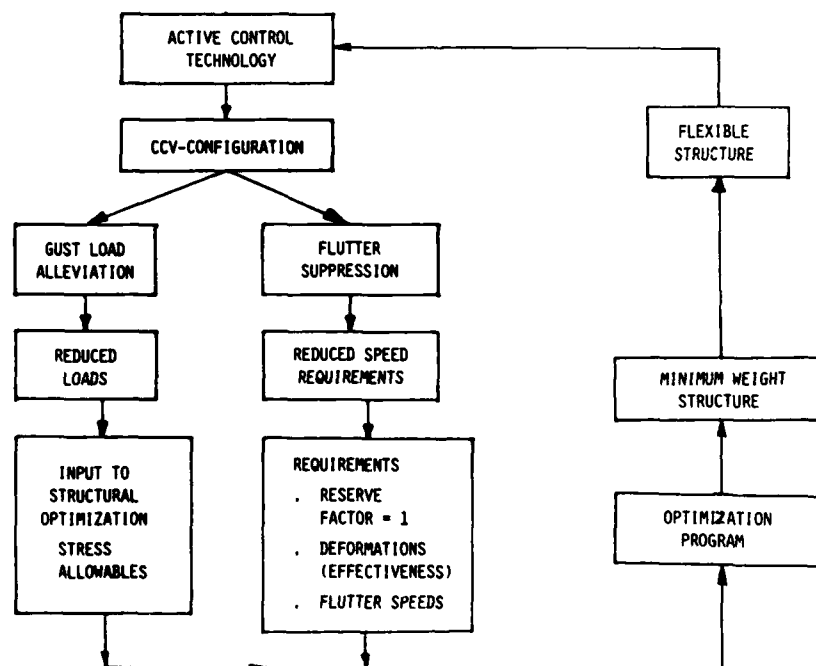


FIG. 6 NEW TECHNIQUES OF RECENT AIRCRAFTS

In addition to stress, displacement, aeroelastic and dynamic constraints an integrated design involves all these techniques and the optimization procedures must be extended for these new constraints.

#### 4. STRUCTURAL OPTIMIZATION AT MBB

After several years of using and modifying some optimization programs (FASTOP, TSO), MBB is now developing an own structural optimization system called

M B B - L A G R A N G E

Fig. 7 shows the intended potential performance of this new program system.

##### 1. REQUIREMENTS

FINITE ELEMENT STRUCTURE

##### 2. STRUCTURE VARIABLES $x \in \mathbb{R}^m$

- o SKIN THICKNESS
- o BALANCE MASSES
- o FIBRE DIRECTIONS
- o NODE CO-ORDINATES

##### 3. CONSTRAINTS $g_j(x) \geq 0, j \in J$

- o MIN/MAX - VARIABLE
- o STRESSES
- o STRAINS
- o DEFORMATIONS
- o FLUTTER SPEED
- o DIVERGENCE SPEED
- o AEROELASTIC EFFICIENCIES
- o EIGEN FREQUENCIES
- o ELEMENT STABILITY
- o SYSTEM STABILITY
- o DYNAMIC RESPONSE
- o WEIGHT

##### 4. MULTIOBJECTIVE FUNCTION $f(x) \rightarrow \text{Min.}$

VECTOR OPTIMIZATION = "TRADE OFF" STUDIES OF  
CONVEX COMBINATION OF OBJECTIVES

FIG. 7 INTENDED POTENTIAL PERFORMANCE OF PROGRAM SYSTEM LAGRANGE

The basic considerations designing a new optimization procedure are listed in Fig. 8. This figure shows the most accepted trends in the development of structural optimization systems.

- o SEVERAL OPTIMIZATION METHODS FOR COMBINING THEM DEPENDENT ON THE PHYSICAL PROBLEM, CONSTRAINTS ...
- o ENGINEERING MODULUS FOR SOLVING SYSTEM EQUATIONS
- o IMPROVE FINITE ELEMENT LIBRARY ESPECIALLY FOR COMPOSITE MATERIALS
- o MODUL FOR QUASIANALYTICAL SENSITIVITY ANALYSIS AND METHODS FOR CALCULATION OF NUMERICAL GRADIENTS
- o ARCHITECTURE FOR HIGH MODULARITY TAKING IN ACCOUNT THE CAPABILITIES OF SUPERCOMPUTER, USE SCIENTIFIC DATABASE FOR MODULARITY

FIG. 8 TRENDS IN THE DEVELOPMENT OF STRUCTURAL OPTIMIZATION SYSTEMS

### 5. CPU TIME EXPENSE IN AEROELASTICS

Two general equations for the solution of the static aeroelastic problem are given in Fig. 9. Solving the equation for the displacement derivatives the big amount of CPU time is mainly demanded by the large number of design variables and also by the number of iteration steps. In Fig. 10 a formula for the estimation of the total number of operations is depicted. An estimation for an example with reasonable size provides  $2 \cdot 10^{10}$  operations, that is a CPU time of about 5 hours on an IBM computer for one optimization step, of course. An approach using the mentioned formulae for the example in question would result in a prohibitive cost.

$$\begin{aligned} \text{I)} \quad & \boxed{K(t) \cdot X(t) = A \cdot X(t) + k} \quad (\text{EQUILIBRIUM}) \\ \text{II)} \quad & \boxed{K(t) \frac{\partial X(t)}{\partial t_i} = A \frac{\partial X(t)}{\partial t_i} - \left[ \frac{\partial}{\partial t_i} K(t) \right] X(t)} \quad (\text{DERIVATIVES}) \end{aligned}$$

K	:	STIFFNESS MATRIX	SOLUTION OF I AND II ITERATIVELY
A	:	AERODYNAMIC MATRIX	
X	:	DEFORMATION VECTOR	
k	:	LOAD VECTOR	
t	:	DESIGN VARIABLE	
i	:	INDEX OF DESIGN VARIABLE	

FIG. 9 SENSITIVITY ANALYSIS FOR STATIC AEROELASTIC

$$N \approx 0.2nb^2 + lms(a^2 + 1.25nb)$$

N	:	TOTAL NUMBER OF OPERATIONS (1 MULTIPLIKATION + 1 ADDITION + 1 STORE)
n	:	DEGREES OF FREEDOM OF THE STRUCTURE (2000)
b	:	BANDWIDTH OF STIFFNESS MATRIX (400)
a	:	AERODYNAMIC ELEMENTS (1000)
l	:	NUMBER OF LOAD CONDITIONS (1)
m	:	DESIGN VARIABLES (500)
s	:	ITERATION STEPS (20)

$$N_{(\text{EXAMPLE})} \approx 2 \cdot 10^{10} \text{ OPERATIONS (ABOUT 5 HOURS CPU ON IBM)}$$

FIG. 10 OPERATION-COUNTING FOR THE ABOVE SENSITIVITY ANALYSIS

A reasonable way to save computer time in structural optimization is to regard some general rules which are given below, but the success would be rather modest.

- o reduce number of degree of freedoms
- o reduce number of design variables
- o reduce number of aerodynamic elements
- o optimize band width of stiffness matrix
- o optimize iterative solution

A much higher decrease of CPU time can be expected by using certain optimization methods for which very fast procedures for the sensitivity analyses exist. These proce-

dures exploit the special characteristics of the optimization methods in mind. Nevertheless, the reductions and optimizations mentioned above should be done wherever it is possible.

In the future, structural optimization will be done using high speed supercomputer with big memory capability.

#### 6. STRUCTURAL OPTIMIZATION USING SUPERCOMPUTERS

An important point in view of structural optimization is the performance of today's and future supercomputers. Fig. 11 shows characteristic data of some well-known computers. The MFLOP rates in this figure are highly theoretical. In practice, however, an average of 10 to 30% of the speed values seems to be realistic.

TYPE	MEMORY (M WORD)	THEORETICAL (M FLOPS) SPEED	CLOCK (N SEC) CYCLE	ADDITIONAL MEMORY
IBM 3084	8	8	26	NO
SIEMENS VP 200	32	536	7.5	...
CRAY-X-MP 48	8	1260	9.5	YES
CDC CYBER 205	16	800	20	YES

FIG. 11

The ratio of wall clock times of scalar mainframe to supercomputer is like hours to minutes. The performance of future supercomputers available in a few years is shown in Fig. 12.

TYPE	MEMORY (M WORD) 8 BYTE	THEORETICAL (M FLOPS) SPEED	CLOCK (N SEC) CYCLE
CRAY-2	256	2000	4
ETA 10	128	10.000	4

FIG. 12

Some aspects are given in the following for the design of a structural optimization system with special respect to supercomputer.

- o Extreme performance by vectorization, large physical memory and multitasking demand for a very new design of the structural optimization system using the above possibilities in a special manner.
- o Minimize wall clock time
  - make a design to maximize vectorization
  - minimize scalar CPU time (especially in finite element codes)
  - make an efficient data management to minimize I/O wait time.
- o In core solutions will become more important.
- o Vectorization and the possibility of virtual memory may be contradictory.
- o Recomputing may be some times more efficient than any data transfer.
- o An adjusted data base is necessary for the optimization of different types of structures. Requirements for a data base should be such as.
  - logical independence of data
  - independent of computer type
  - FORTRAN interface
  - variable record length
  - handling of extensive data (matrices, vectors)
  - multi user operation



7. REFERENCES

- /1/ GÖDEL, H.; SCHNEIDER, G.; HÖRNLEIN, H.  
Preliminary Design of Aircraft using Structural  
Optimization Methods  
AGARD-Conference Proceedings N. 354, London 1983
- /2/ SCHMIT, L.A.; FLEURY, G.  
Genesis and Development  
AIAA-Journal, Vol. 19, No. 10, 1981
- /3/ VANDERPLAATS, G.N.  
Structural Optimization - Past, Present, and Future  
AIAA-Journal, Vol. 20, No. 7, 1981
- /4/ VENKAYYA, V.B.  
Structural Optimization: A Review and some Recommendations  
Int. Journal for Num. Math. in Engr.  
Vol. 13, pp. 203-228, 1978
- /5/ ASHLEY, H.  
On Making Things the Best - Aeronautical Uses of Optimization  
Journal of Aircraft, Vol. 19, No. 1, 1982
- /6/ FLEURY, G.  
Reconciliation of Mathematical Programming and Optimality Criteria  
Methods  
Chapter 10 from "Foundations of Structural Optimization" edited by  
A.J. Morris, John Wiley & Sons, 1982
- /7/ HAUG, E.J.; ARORA, J.S.  
Applied Optimal Design/Mechanical and Structural Systems  
John Wiley & Sons, 1979
- /8/ MORRIS, A.J.  
Foundations of Structural Optimization: A Unified Approach  
John Wiley & Sons, 1982
- /9/ SIDDALL, J.N.  
Optimal Engineering Design  
Marcel Dekker Inc. 1982
- /10/ REKLAITIS, G.V.; RAVINDRAN, A.; RAGSDALL, K.M.  
Engineering Optimization Methods and Applications  
John Wiley & Sons, 1983
- /11/ HAFTKA, R.; KAMAT, M.  
Elements of Structural Optimization  
Martinus Nijhoff Publishers 1985

## THE EFFECTIVE USE OF COMPUTING POWER IN STRUCTURAL ANALYSIS

by Ian C. Taig  
British Aerospace P.L.C., Aircraft Group - Warton Division  
Warton Aerodrome, Preston PR4 1AX England.

### SUMMARY

This paper supplements earlier surveys of the potential structural applications of large scale computing by introducing two factors not previously considered. These are a major conceptual change in the task of modelling for finite element analysis, made feasible by cost-effective large number-crunching capacity, and the emergence of expert systems techniques to broaden the scope and increase the effectiveness of large scale computing.

These two factors allied to the data handling and storage needed to manage them, can lead to demands for two to three orders of magnitude increases in computing power in the next decade. They may also influence the computer hardware configurations required.

### 1. INTRODUCTION

In most previous surveys of large-scale computing (e.g. in the recent report (1) to the Fluid Dynamics Panel), the trends in structural computing are presented as fairly straightforward extrapolations from today's requirements. This leads to an inevitable conclusion that, as regards large-scale computing in aeronautics, it is mainly fluid mechanics which will set the demands while structures will use up (hopefully effectively) some of the capacity made available to meet these demands. This is because structural computing using FE methods has reached a temporary plateau where MOST of the things deemed ESSENTIAL can already be done - at a cost and in a timescale which industry has come to accept as the norm. Further advances, through the availability of more and cheaper number-crunching, storage and data handling are seen as contributing to larger and more extensive optimisations, more iterative cycles and hence more non-linear capability, longer and more representative dynamic analyses and so on.

This view can be summarised in the grossly oversimplified statement:

"Further advances in structural computing represent increments on a curve of diminishing returns, which are justified mainly by the improved cost-effectiveness of computing power, rather than by necessary technical advance".

Whilst there is a large element of truth in this view, it overlooks two facts which may have a very significant bearing on how we analyse and design structures; we may not simply be looking for "more of the same, but a lot cheaper".

The first is that, as capability to analyse in greater detail increases, our conception of the crucial task of structure modelling could undergo a radical change. At present the practical scale of analysis (dictated by the ability to prepare, validate and handle the data as much as by central computing capacity) falls far short of the "ideal" in which local and global behaviours can be handled by a continuous set of calculations. We refer to this as the question of "scale compatibility".

The second important fact is that the finite element "black box" analysis packages, now universally available, provide as much capability for incompetent and ineffective analyses as for effective use. Poor results can be produced faster, larger and larger models created, whether necessary or not, by the mesh generators and modellers. The new emerging fact is that we are learning how to inject not only "intelligence" but also the fruits of experience into computing systems. The new generation of software/hardware tools for analysts and designers will significantly alter the computing balance between solving the numerical/mathematical problem on the one hand and formulation and interpretation (in real world terms) on the other. We refer to this as "assisted utilisation".

These two issues will be considered before returning to the question of structural demands for large-scale computing.

### 2. SCALE COMPATIBILITY AND A NEW VIEW OF MODELLING

In airframe structures it is not yet feasible to handle, in a single, coherent set of analyses, all structural problems from global stress/displacement distribution down to stress concentration around a crack or in the individual layers of a laminate. This is one essential difference between structural and fluid mechanics analyses, which we can summarise by comparing the scale of significant physical behaviour with the scale of geometric features which cause that behaviour. In fluid mechanics, representation of the 'determining geometry' is a relatively minor problem - the physical phenomena themselves need modelling down to a scale (in space and time) which is far more demanding. In structures, it is at the limits of current computing power just to describe the geometry down to the last significant detail and one to three orders of magnitude beyond these limits to produce a total and coherent set of analyses spanning the whole scale range. We can currently only manage a two or three - stage process in which the truly local analysis is concentrated in areas identified as critical by a coarser, higher level

study or by test. Even sub-structure analysis, which is meant to bridge this gap, rapidly becomes too unwieldy to be managed by any but the most expert analysts.

A simple extrapolation of current trends shows that this situation could change within the next five years, and certainly within a decade. If we once pass the "manageability" barrier a whole new (for aerospace) concept of structural modelling becomes feasible. In the past it has always been necessary to use a considerable degree of skill and judgment in deciding how (or even whether) to represent fine detail within the bounds of individual elements. Skilful modelling can be the most time-consuming of all data-preparation processes; unskilled modelling can destroy the value of the most complex analysis. In the limit, assuming that the hardware/software aids are sufficient to create and manage all the data, modelling as a task disappears - we simply include every feature explicitly by defining a standard mesh representing its geometry, and by assigning realistic material properties to every tiny element so created.

To give a rough scale to this approach, consider the analysis of a conventional aircraft wing. In a typical present day case we might use a mesh with 20 - 40 span-wise stations, 5 - 20 chord-wise and 2 - 6 depthwise for a single - pass linear static calculation, supported by up to 20 local analyses in identified problem areas and as many more analyses of associated components such as flaps and control surfaces. The largest analysis is typically about 1000 nodes, 3 - 5000 degrees of freedom. (Fuselage analyses may be 5 - 10 times this size). The total set of wing analysis tasks could be 10 to 50 times as large as the typical 1000 node basic analysis. This is easily handled on today's serial-processing computers.

Consider detailed representation in two stages:-

- a) representation of reinforcement (stiffeners and/or lamination graduations), and substantial features (access holes, doors, panels, mounting etc.)
- b) representation of individual joints/fasteners and their associated local structural changes, individual laminae or lamina groups.

At level a) we should seek a representation scale which at least permits local buckling to be investigated in the main analysis set. At level b) we represent all features in sufficient detail to identify characteristic stresses which can be associated with specific failure criteria.

Representation a) will result in a mesh size of the order of 200 length wise x 50 chord wise x (on average) 10 depthwise, i.e.  $10^5$  nodes - 2 orders of magnitude beyond current practice and still needing subsidiary calculations in support.

Representation b) will demand the same basic mesh with refinement to 1/10th of the linear scale in discrete regions. This might mean 5 - 10 times the total size. We may use this as a convenient measure of total analysis content in both cases.

One-off analyses of up to  $5 \times 10^5$  d.o.f. have been performed, using multi-level sub-structuring, on present day computers so there is no question about the feasibility of analyses up to  $5 \times 10^6$  d.o.f. using the best Class 6 machines. For automated modelling we shall need computer-based solid geometry to the same level of detail and it may be geometry generation, storage and manipulation which will determine computer capacity. The hardware/software advances expected in the next 10 years should certainly bring the  $10^6$ - $10^7$  d.o.f. super-analysis into the practical range, if we decide that this is what we need and develop the tools to achieve the goal.

At the cost of a very large development effort to produce a balanced set of tools for generation, handling, storage and display of the vast amounts of data - but with the prospect of greatly reduced time and effort in analysis preparation and interpretation - a totally different picture of structural computing demand would emerge. This scenario demands high usage of the best foreseeable computing equipment - the whole philosophy collapses without the quantum leap.

### 3. ASSISTED UTILISATION OF STRUCTURAL COMPUTING TOOLS

It is a banal truth that a structural expert using simple and modest computing tools could always out-perform an idiot using the best, (even idiot proof) available tools. The ingrained appreciation of "what actually matters" and the ability to read across significant information between global and local analyses, omitting the irrelevant, adds an extra dimension to the potential effectiveness of simple tools. It is equally true that people of the calibre needed to achieve this top level performance do not, and cannot be persuaded to, do such a job for any length of time. We do not think it feasible to postulate 'genius-level' performance from the day-to-day analysts as a viable alternative to the super-analysis previously considered.

However an improved, and self-improving level of performance can potentially be obtained from proficient, but non-expert analysts if we introduce expert-systems concepts into data preparation and output interpretation stages of finite element analysis. At present, we are only learning the first principles of constructing expert systems - how to capture the knowledge of experts, store it and make it accessible to users at the time of analysis preparation or interpretation. The tools we are using are clumsy and incomplete and it has not yet been shown to be feasible to construct a really large and complex system such as might be required to provide a general purpose, "intelligent" front end to a large finite element analysis suite. However, we have reached the stage

where individual steps along that route have been taken. A prototype system SACON (2) has been available for some years from the U.S.A., it is very limited in the problems which it can address: concerning itself with the type of analysis to be performed rather than with the larger problems of modelling and data preparation. Two U.K. system studies are tackling these problem areas. NASCON (3) is being developed to aid the inexperienced user (who knows what problem he wishes to solve) to select the appropriate data entry and solution facilities for solving his problem using NASTRAN, with explanation and help tailored to his needs. Our own FEASA project is studying the problem of helping the user who is unfamiliar with the details of finite element analysis to formulate his "real world" problem properly and make appropriate decisions concerning mesh shape and size, element type selection, solution procedures, constraint representation etc. - the first level decisions in modelling.

These studies lead to the firm conclusion that the knowledge concerned can be transferred by this route - but with some difficulty in its formalisation and at a high cost in the size and complexity of a practical computing system. For example, the FEASA system already involves over 1000 actions and derivations (the entities used in its SAVOIR shell system for describing knowledge) and uses a large part of the capacity of a VAX 11/750 with barely adequate response time. To perform a useful general purpose job it would need to be expanded to about three times its present size. At this size, it might only be addressing about a quarter of the user interface problems involved in a simple linear static analysis and less than 10% of those facing all users of a large FE system. The computing capacity to support an "intelligent interface" with a large FE system with many simultaneous users is thus certain to be significant and may be very large. Furthermore, this capacity is being deployed in highly interactive dialogue with users and will rapidly clog up communications if the computing power is moved far from the user.

Present day technology suggests that the ideal way to introduce intelligent interface systems is to run them on high power super-micro work stations with multi-windowing. The support system, using text and elementary sketch-pad graphics is displayed alongside, and shares the physical problem data base with, the input data and/or the solution output. Solution of the analysis can be handled on the micro itself, on a departmental, large mini or on a corporate large mainframe, according to the job size. The pace of development of packaged power suggests that the departmental mini-computer may become redundant in the evolution of an integrated corporate computing system.

The relevance of this discourse to large-scale computing, in the narrow sense, may not be immediately obvious; but the provision of expertise at the user's disposal at the work place is certain to build up a more interesting and involved dialogue which will probably lead to an interest in the problem being solved and a desire, on the user's part, to be in control of the modelling and interpretation rather than a reliance on remote number crunching. It could thus be seen as a "human alternative" to the remote solution, whose very vastness takes it out of the user's hands. It is not, however, necessarily such an alternative, since the expert interface is as usable for organisation of the large problem as it is for intelligent modelling at the smaller scale. It might lead to a different view of the disposition of computing resources, requiring more power at the user's elbow and probably less at the centre; it therefore affects priorities, investment and development funding.

The introduction of the expert systems concept into structural analysis changes the black-and-white nature of the decision to go for explicit detailed modelling and the absolute necessity for vast computing power which goes with that decision. It also helps arrest the spiral descent into mediocrity on the analyst's part: the self-fulfilling prophecy which is fundamental to the case for ultra-fine detail.

#### 4. EFFECTS ON COMPUTING REQUIREMENTS

Combining the above considerations with the earlier extrapolations of structural computing needs suggests a more positive requirement for order-of-magnitude increases in computing power than have previously been suggested. Table 1 provides a simple summary. There is an increasing trend to use computer-based analysis, supported by detailed analysis and testing as an alternative route to airworthiness clearance in place of full scale testing. The requirement for increased confidence in the accuracy and reliability of analysis is growing. At the same time, the typical analysis user is likely to be a 'slick' computer key-board practitioner and a reasonably trained structural engineer, but is most unlikely to be expert in the fundamentals of finite element analysis.

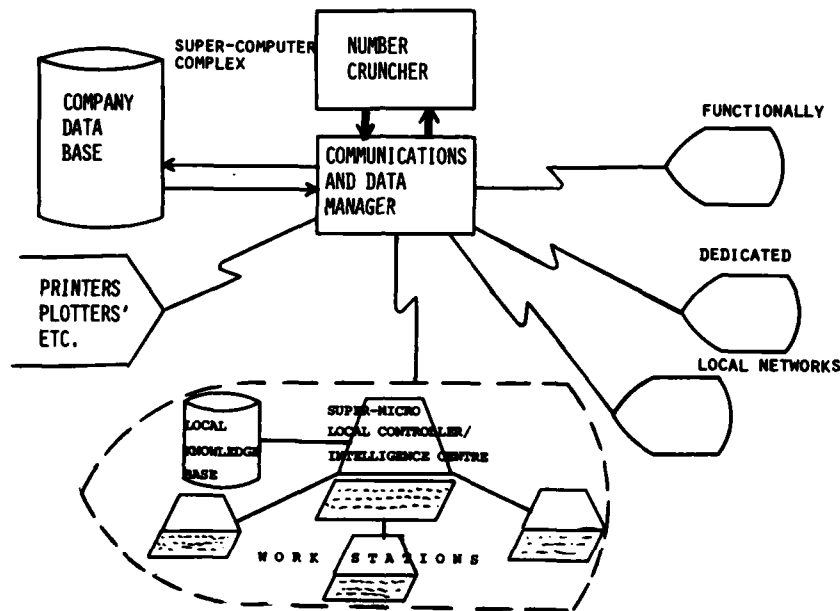
We see the provision of tools to support the analyst, either by explicit modelling and vast numerical simulation or by expert assistance at data preparation and interpretation time, not as optional 'frills' but as essential elements in maintaining standards of integrity and improving user productivity. In the writer's judgment, the provision of the self-tutoring expert assistant is marginally the more necessary, because it is safer to ensure improved understanding of the analysis task than to rely on a process which largely by-passes human judgment.

Developing the reliable software needed to support both approaches is a substantial task requiring the involvement of aircraft industry finite element specialists as well as analyst programmers (who need not come from the industry itself). We see a need, in both cases, for software tailored to the special needs of the aircraft industry but strongly support the use of more general-purpose software 'tool kits' and data base management systems in order to produce compatible systems by design rather than by adaptation.

Software development timescales for both types of system are similar:- 3-5 years for evolution of reliable tools for general design use. This suggests that we should aim to develop from our present style of computing (large mainframes plus distributed, functionally specialised minis plus more-or-less intelligent terminals), via access to a large super-computer for system development, towards an eventual system in which a large central number-crunching and data management facility is coupled as directly as possible to highly intelligent and functionally specialised workstations. Our ideal solution will provide both kinds of computing environment: the total geometrical/numerical structural modeller and the expert-assisted interface.

FIG. 1

## IDEALISED COMPANY COMPUTING FACILITY



The ideal computer installation suggested in the grossly simplified figure 1 is not unlike today's arrangements except that the role of the specialised mini computer network is replaced by more centralisation of corporate data management and CPU power and very powerful microcomputers in small local area networks. We see the timescale for migration to this type of facility as about 5 years, i.e. aiming for 1990, dictated largely by software; with an interim period, starting in 1986 at the latest, where access to a software-compatible super-computer is necessary.

### 5. REFERENCES

1. AGARD-AR-209 "Report of the Working Group on Large Scale Computing in Aeronautics". Report to FDP June 1984.
2. Stanford University Report HPP-78-23 "SACON: A Knowledge-based Consultant for Structural Analysis" ARPA Contract MDA 903-77-C-0322. J. Bennett, L. Creasey, R. Englemore, R. Melosh September 1978.
3. Marconi Report ITM 84/74 "NASCON - A Knowledge-based Consultant for NASTRAN" R.A. Corlett, A.T. Humphrey, J.N. Trumpeter Presented to NASTRAN European Users' Conference May 1984.

### 6. ACKNOWLEDGEMENT AND DISCLAIMER

The writer wishes to thank British Aerospace p.l.c. for permission to present this paper as a discussion document. The views expressed are the author's and do not represent the considered opinion of the Company.

TABLE 1

STRUCTURAL COMPUTING REQUIREMENTS SUMMARY

For a large aircraft company by 1990

Type of Capability	Scale Indication	Urgency of Requirement
Basic finite element analysis.....	Models and solutions up to $10^7$ d.o.f. ( $10^2 - 10^3 \times$ IBM 3081K capacity)	Desirable for productivity through full automation
... with ultimate data links to.....		
... Detailed solid modelling of airframe geometry.....	Detail definition at approx. $10^{-4}$ x linear dimensions in 3-d solid model	Essential as accompaniment to the above
... with design links to...		
... Optimisation	$10^4$ to $10^5$ variables 100 simultaneous design conditions	Highly desirable for Competitive edge
Non linear analysis	$10^3 - 10^4$ d.o.f. $10^2 - 10^3$ iterations	Desirable
Intelligent front-end	At least VAX 11/780 power dedicated to support each 6-10 users - sharing FE analysis database	Highly desirable/necessary for safe use of complex 'black box' systems

Computer Technology and Engineering  
-Past, Present, and 1995-

N. A. Radovcich, Ph.D.  
Research and Development Engineer  
Flutter and Dynamics Department  
Lockheed-California Company  
P.O. Box 551, Burbank, CA 91520 USA

## SUMMARY

Significant investments in computer software and hardware support engineering design, analysis, and synthesis functions. Updating these investments to accommodate changes in hardware and software is a major concern to engineering.

The forces shaping the engineering computing environment in the next ten years will be dynamic and multifaceted. This paper discusses the role of the more prominent forces and suggests ways to focus these forces on satisfying engineering computing needs.

## LIST OF ACRONYMS AND DEFINITIONS

AT&T	American Telephone and Telegraph
bit	a binary element/line/pin
byte	contains eight bits
CAD/CAM	computer aided design/computer aided manufacturing
CDC	Control Data Corporation
cpu	central processing unit
DEC	Digital Equipment Corporation
DG	Data General Corporation
FAMAS	Lockheed's matrix data based computing system for aeroelastic analysis
Giga ...	-10**9 ... such as in Gbytes or gigabytes
IBM	International Business Machines Corporation
IBM PC	IBM personal computer which uses Intel 8088
i/o	input/output
Mega ...	10**6 ... such as in megabytes or megabits
MegaFLOPS	millions floating point operations per second
M680xx	Motorola's complete 32-bit microprocessor family
MC68020	Motorola's 16/32-bit microprocessor family
Mhz	Megahertz, usually the clock frequency units of a microprocessor chip
MIPS	millions of instructions per second
NS32032	National Semiconductor complete 32-bit microprocessor
NASA	National Aeronautics and Space Administration
NASTRAN	structural finite element program developed by NASA
UNIX	a multiuser, multitasking operating system; is a trademark of AT&T Bell Laboratories

## INTRODUCTION

Computer technology today retains many aspects of the pioneer spirit and the excitement of an industry whose horizons are unlimited.

While the pioneering excitement continues today in artificial intelligence, robotics, speech recognition, and real-time processing, an equally enthusiastic segment of the aerospace community desires to integrate matured engineering design processes into a cost-effective, computer-aided design tool. Significant investments in computer software and hardware support engineering design, analysis, and synthesis functions. Updating these investments to accommodate changes in hardware and software is a major concern to engineering.

Of equal importance is the integration of advanced design capabilities such as non-linear aerodynamic-structural interaction codes, structural sizing with active controls, and battle damage tolerant system analyses into the design process. The goal is to increase the engineering design capability while controlling the cost of improving the computing environment.

The forces shaping the engineering computing environment in the next ten years will be dynamic and multifaceted. The objectives of this paper are to identify and understand these forces, to convert perceived liabilities into assets, and to focus hardware and software advancements toward satisfying future engineering computing needs.

## PROBLEM DEFINITION

One of the most controversial subjects in engineering is the apparent uncontrolled increases in engineering computing costs associated with high-technology designs. The controversy starts with identification of the cause of the high computing costs.

In a hypothetical structural analysis and synthesis effort, it can be argued that a finite element structural model with twenty thousand degrees of freedom and the associated two thousand load cases is one of the principle causes of the increases in computing

costs. In defense, the structural analyst points to the design requirements for an accurate representation of the physical system demanded by the structural guarantees which are in the contract with the customer.

The analyst argues that if the computing environment were improved, the total cost of the design would decrease. He identifies the computing equipment and the software as being the primary problem areas.

The computing division counters that it is given a fixed budget to service engineering computing needs, and that computing technology is changing so fast that equipment changes must be optimized to minimize computing costs.

The project program manager hears words such as synergisms and integrated design efforts. The tradeoff between fighting for a step increase in computing capacity to translate these advanced concepts into actual practice, and staying with past design concepts and available computing capacity is a difficult no-win decision. This first option will incur a step increase in cash-flow, while the second option will probably incur schedule slipping and design requirement compromises.

There are no simple solutions to the above conflict of computer resource allocations. The following sections will discuss the forces that have formed the engineering computing environment during the past twenty years. An approach to satisfying future engineering computing needs will then be proposed within the constraints implied by the understanding of the dominant forces governing the future hardware and software technology developments.

## COMPUTING ENVIRONMENT

### User Profile

The engineering computing environment is a complex interrelationship of user access to:

- 1) interactive processing for data preparation and review of output data
- 2) cpu and i/o engines
- 3) flexible and comprehensive computer operating systems
- 4) print/plot/microfiche/etc. forms of output
- 5) database management systems
- 6) proven computer programs
- 7) high-level system's integrator
- 8) software development and software support
- 9) documentation
- 10) capital resources to afford items 1 through 9.

An inadequate computing environment is equivalent to an ill-defined mathematical statement. To the engineer, both are barriers to a solution. In a deficient computing environment, the engineer will go beyond the user role and do whatever his organization will permit to resolve the deficiency; for example, develop software, buy equipment, buy timeshare services, etc.

### Computers and Organizations

The computer is the modern day two-edged sword. This high technology tool can either increase the organization's productivity or lead the organization into serious difficulties. Sometimes a manager credits the computer with doing both simultaneously.

Computers are a vital resource in a technology-based organization. Those who control this resource are close to inter-company power struggles. Computing resources, therefore, are the favorite food of empire builders, cost cutters, endless justification requesters, buck passers, user tax collectors, soft-dollar versus hard-dollar debaters, and tax write off pushers. Computer facilities create internal activities apart from the primary goals of the organization. In most technology-based organizations, allocation of computer resources resembles more an art form than a standard business practice.

Organizations classify computers both as utilities, such as telephones, and as luxuries, such as excessive wage demands. Merging the requirements of the "the bit and byte" user, the number-cruncher user, the business user, the computer-knowledgeable user, the interactive user, the graphics user, and the operating system manipulator user into one facility challenges the imagination of the most optimistic neophyte. What is a luxury to one user is a necessity to another.

There are two purposes for computer installations: to increase the productivity of the organization and to provide the organization with the capability to function properly.

Problems arise when the organization attempts to keep computer costs to a minimum and ignores the total cost of doing business. Computing facilities grew to provide technical



expertise in making computers and software serve the customer in the most cost-effective manner. Computer facility organizations became computer resource power brokers. A good data processing manager shows high load factors on the computing machines.

How the computer facility policies affect the overall cost of doing business is either ignored or considered too complex to include in facility cost tradeoff studies. These barriers are coming down, but the origins of the data processing mentality persist.

#### Competing for computer resources

A company generates a plan, an organization, etc., to accomplish its objectives. This process then translates the company resources into specific objectives. Figure 1 shows a simplified model of the company function. If the company objective is profit, the process uses facilities, material, money, and people to produce that economic return. Included in the company objectives are day-to-day operations, such as general accounting functions. If the computer is in the resource box (Figure 1), there is a clear competition among different processors in the process box to access and use that resource. If the computer is located in the process box, then the competition for the company resource is at the money level. The computer has a dual personality relative to the resource and process identities in most companies.

Computing environments derived from a least-square fit of the organizational computing requirements usually cannot satisfy the specific engineering requirements. Monolithic computer facilities are tuned to service the major customer. Engineering analysis is rarely a major customer of computer resources, if only because of the cyclic nature of design.

The engineer uses many tools to translate an idea into a practical reality. A design must satisfy the economics of the marketplace as well as the functional requirements. The engineer is a composite of scientist, lawyer, and entrepreneur. The design process includes senior judgement, science, planning, designing, manufacturing, and marketing.

Whether the computing environment is excellent or poor, the engineer will design. The better, more cost effective design will depend on the statement of work, the organization cost accounting practices, personnel, and schedule requirements. An enlightened and informed organization knows where increased computing costs will reduce overall product costs.

#### On computing technology developments

Advancing technologies in hardware and software pose an unusual dilemma for the engineer. If new engineering design products are ignored, the organization becomes technically obsolete. If the new products are incorporated into the design process as soon as they become available, the organization could use all of its resources merely trying to be on top of the high-tech curve.

Organizations that have nonchanging technology requirements for productivity increases, or high output of high-end technology products can survive these extremes. The situation becomes interesting when there are organizations at the extremes within the same company. The problem becomes impossible if the requirements of many companies within the corporate structure are integrated into a single policy or computer-use definition.

The computer industry makes pronouncements of great technology break-through, and expectations rise in user communities. Translation of the breakthrough into reliable and

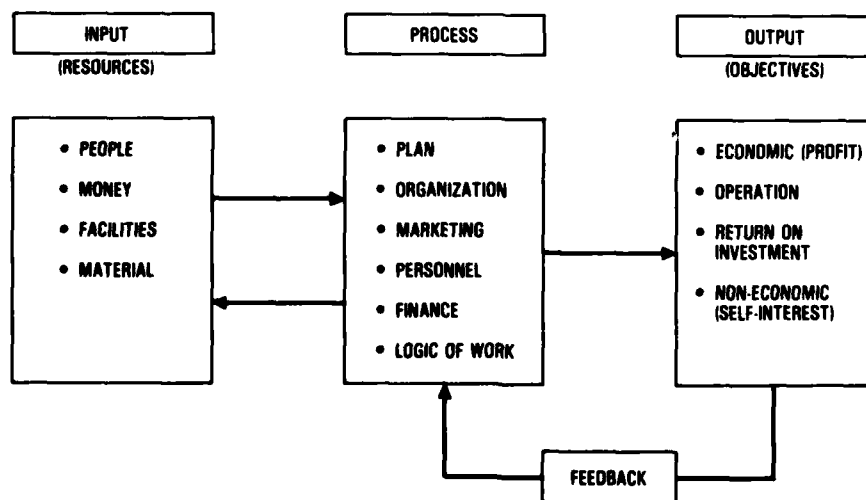


FIGURE 1. COMPANY FUNCTION

economical hardware and software does not always happen. Even when technology barriers have been successfully scaled and manufacturing is feasible, marketing may decide that the new product would undercut one of the more profitable lines and therefore delay its entry. Projections on the future computing environments which were based on the computer industry pronouncements are not feasible.

#### Role of management

Of all the issues which determine the computing environment for engineering, the role of management is the most difficult to evaluate. The user knows the computing requirements and how to interact with the computer. Finance wants to allocate minimum resources to accomplish the organization's goals. The computer industry wants to sell the newest computer technology. In the middle of the foray appears management.

R. Richard Ritti in The Ropes to Skip and The Ropes to Know, says that the fundamental truth known to seasoned managers is the "...best decision is no decision at all." When the manager has complete information, the "...task is simply to pick the best alternative." This good decision is no decision at all. The second situation is when the "...the possible outcomes are known with probabilities attached to these outcomes." Again, the manager picks the best one, usually the one with the minimum risks and the maximum benefits. Finally there is that situation which is filled with uncertainty. The manager's gameplan, then, is to find someone to "...take the risk while making sure that it is he who stands to gain." The manager "...will leave the decisions to someone else."

The acquisition of computer resources must involve one of the best played games at making no decisions because of the uncertainties and because of the large investment the organization must make. This is a major factor to address when assessing the direction of future computing environment. Contributing to the uncertainties is the lack of managerial tools to accurately predict computing needs in terms of the organization's strategic goals and the effect of reducing the total cost of doing business.

Another consideration is the time required to develop a top engineering manager versus the time required for the computer industry to bring to the market hardware or software which radically changes the computing environment. A top manager today who was fortunate to have had any computer hands-on experience was probably working with a deck of input cards and magnetic tapes.

What are the top engineering managers' perspectives on attaching monitors, relational database managers, front-end computers, terminals on the desk, access to printers and plotters, and computing power measured in 10s of megaFLOPS (millions floating point operations per second)? How do these topics relate to the technology competitiveness and productivity within engineering? How do these topics relate to keeping within budget and schedule commitments?

#### COMPUTER DEVELOPMENT FORCES

##### From humble beginnings

In 1958, the author visited the germanium diode computer installation at Point Mugu Naval Air Station. A special vault isolated the computer from exterior electromagnetic interference and kept the installation within 1 degree of the designed operating temperature. An elaborate environmental control system monitored the humidity as well as the temperature. Fire alarm bells rang when tolerances were exceeded.

These first digital computers were temperamental and extremely limited in computing power; however, they provided important operational data and user interface experience. Universities used these first-generation computers to help develop courses in computer science. Today, the universities are following a similar path with the first CAD/CAM computer systems.

Structural design numerical problems were probably one of the first uses of commercially available digital computers such as the IBM 704. These first machines had limited computational capabilities: no input/output devices except for a teletype system terminal, a card reader, a card output punch, and a printer. They were under engineering control, and the first operators were engineers. While the machines were slow relative to current computers and difficult to keep running, their impact on engineering problem solving capability were felt immediately. The computer performed numerical computations in one day that took weeks for a team of people on mechanical calculators.

The key element in mathematical models for structural and dynamic analyses is the number of degrees of freedom (DOF) which is permitted in the problem formulation. The first digital computers, when compared to a team of people on mechanical calculators, were cost effective, with increases in DOF of two to three times. The digital computer economics for these types of engineering problems initiated the trend of justifying higher computer unit pricing for lower solution costs per problem.

The performance to price ratios quoted by computer manufacturers is today a continuing source of conflict for engineering management. During low design activity, the number of problems is small and the cost per solution is large. During periods of high design activity, the cost per solution is low. However, the high computer turn-around times can stretch out schedules and cause high total task costs. To engineering management, computers are a no-win proposition.

Today, the digital computer has made major inroads into banking, manufacturing, airline reservations, payroll and accounting, artificial intelligence, expert systems, word

processing, defense systems, robotics, energy distribution networks, the charge card industry, online inventory and control, the stock market, telecommunications, workstations, and all aspects of engineering design and testing. Digital computers are found in security systems, aircraft and spacecraft flight control systems, tanks, cars, trucks, trains, missiles, ships, appliances, etc.

The computer today influences every aspect of a company/government installation, not only in high-end technology operations, but in day-to-day functions such as word processing and information management. The trend today is real-time applications for planning, budgeting, and tracking of project schedules and goals. The computer, which began servicing exclusively engineering computing requirements, is providing a general utility to the whole organization.

#### Computer Marketing Forces

The computer industry is continually having highs and lows. Companies that achieved industry leadership status during their existence have an average life span of less than ten years. A five-year old company, from initial product development to 100+ million sales, talks about 60 percent of the market penetration and contrasts its mature company with the new upstarts. At the other end of the spectrum are the market pacers, who see market penetration and market share as the primary motivation to product developments. If the specialized customers' needs are served also, it is accidentally. The market pacers are experts at making the customer feel privileged to be served by them. Their products are vehicles for their marketing and sales engines. If a new market segment develops, market pacers step in and liquidate the assets. The market pacers rarely make hardware improvements not required by at least 80 percent of their customer base.

In a recent report, Paine Webber Inc. analyst Jonathan M. Fram shows that a \$200-billion-per-year, computer-related industry market exists in just three of its many facets, namely a \$90-billion existing data processing industry, an \$85-billion telecommunications industry, and a \$25-billion workstation industry. PC RETAILING reports that the personal computer industry alone is \$30 billion. The computer industry is the nation's fourth largest advertiser, spending an estimated \$3.1 billion each year.

Meanwhile, the hallmark of the engineering numerical processors, the Cray and CDC 205 (the so-called supercomputers), market share is about 20 machines per year at a cost of \$20 million for each installation. This \$400 million market is pale when compared to the workstation market alone. How is it that the engineering-related numerical calculations which first heralded the power of the computer have been reduced to an insignificant footnote in the computer-market sales activity? Productivity gains in areas other than large-scale engineering problems appear to be the market forces that have the big blue and other computer manufacturers satisfying the customers' needs. While large-scale numerical problems were at one time the technological force for computer development, it is not so today. It is no wonder that engineering finds itself on business machines.

#### Upward push

The most dramatic elements in computer technology development have been in the micro-processor field. Five years ago, the computer market was segmented into the following groups:

<u>Groups</u>	<u>Subgroups</u>
o MAINFRAMES	
super number cruncher, (CRAY, CDC)	supercomputers
number cruncher, (CDC, Univac)	mainframe computers
transaction machine, (IBM)	
o OTHERS	
32 bit processor, (VAX)	supermini computer
16 bit processor, (DG-Nova)	minicomputer
8 bit processor	microcomputer

Today, the 16-bit processor personal computer replaces the 8-bit processor microcomputer with 8-bit and 16-bit data paths. An Intel 8088 16-bit processor running at 8 Mhz with 8087 co-processors is only 10 times slower than the VAX 11/780 supermini computer for a single precision floating point intensive problem.

The M68010 processor is a 32-bit internal processor with a 16-bit data path and 24-bit address capability. This processor is running at 12 Mhz in many computer systems. Finally, a number of full 32-bit processors such as the MC68020 and the NS32032 are currently being implemented on full 32-bit backplane. These processors will run at clock frequencies up to 20 Mhz and will have numerical co-processors.

Traditional mini and supermini markets are experiencing pressure from the microcomputer developments. The supermini suppliers, therefore, are upgrading their performance-to-price ratios to where their software edge will keep the supermicro developments at bay. Supermicros are microcomputers which exceed 0.8 MIPS computing power. A sign that the supermini vendors are leaving the 1 to 3 MIPS machine market to the supermicros is evident in the recent announcements from the supermini vendors.

DEC upgraded its top-of-the-line VAX to compete with the traditional mainframe market. The top-of-the-line VAX 8600 supermini computer replaces the 7-year old VAX 11/780. DEC is offering the 8600 with 4M bytes of memory for 2.4 times the cost of the 11/780 with 2M bytes of memory. However, the 8600 system performance can be as much as 4.2 times that of the 11/780. The manager of the VAX marketing group contends that a VAX cluster comprising seven 8600s, one 11/780 and 40G bytes of disk storage can equal the performance of the IBM 3084 high-end mainframe for 33 percent less cost. This appears to be true if no application running on the system exceeds the bounds on any single processor within that cluster.

The introduction of the 8600 in the 3-5 MIPS offering will start the game of leapfrog between DEC and its major competitors -- Data General, Prime Computer, Perkin-Elmer, Harris, Gould, and Hewlett-Packard. The goal of this game is to produce a general-purpose supermini offering 8 to 10 MIPS by 1988 -- 1990. The offerings in 1995 by extrapolation could be 10 to 15 MIPS. The time has come for a new name, mainframe-minis.

While DEC is emphasizing the corporate data processing market, Convex Computer Corporation (Dallas) is beta testing a mini supercomputer that integrates both scalar and vector processing. The computer is said to run VAX Fortran applications essentially unchanged and yet offers one quarter the processing power of a Cray at one fifth the price. This would put it in the 8600 price class with more numerical computing power than the IBM 3083.

Meanwhile, DEC is facing increasing pressure from IBM's 43XX and the 3083-CX, which is the 308X-family's entry-level processor. IBM has positioned its products to compete with the superminis offerings in performance and price.

#### Focussing on productivity

The computing power of the microprocessor made possible the packaging of the numerical computing power of the VAX 11-780 into a dedicated single-user machine. While performance-to-price ratios were a factor, a single-user machine focussed for the first time on user productivity.

Engineering recognizes that the total project cost is the governing factor to control. Single-user machine costs are well-defined. It does not require a systems support group, on-side hardware maintenance group, special environmental support such as air-conditioning and noise isolation, and complex user use algorithms for accounting purposes. It provides a nondegradable machine response, tailoring of the machine function with minimum contention with other users, portability, and powerful graphics capabilities. The application of this technology made the engineering workstation industry into a major marketing force in less than five years.

#### The shrinking chip

The computer-chip industry produces a large variety of products, including microprocessors, memories, memory controllers, memory management units, floating point units, timing control units, address latch buffers, data buffers, and general support chips. The dominant player among these chips is the microprocessor which usually has a strong association with product line using chip, such as the Intel 8088 in the IBM PC.

The microprocessor comes in a number of models, each sporting different numbers of data, address, accumulator, and processing bits. The power of the microprocessor is measured by the chip clock frequency and the number of clock cycles required to complete a particular function. Some models have floating point co-processors to speed floating-point calculations, symmetrical instruction architecture, instruction prefetch, virtual memory, etc. They offer the system integrator a variety of solutions for assembling computer systems. Each manufacturer of the microprocessors is upgrading its product line toward a full 32-bit processor.

Memory chip development is even more dynamic. Three years ago, computer manufacturers were concerned with whether or not 64K by 1 bit chips were going to be available in commercial quantities. With the delay of the 64K chips, the 256K by 1 bit chips gained prominence and are currently available in mass quantities. AT&T recently announced that it will mass produce a 1 Meg by 1 bit chip. While a chip controller is not currently available, the message to computer integrators is clear. Memory costs will drop dramatically.

An integrated computer board (one-board computer) with 2 Megabytes of on-board memory is available today in the Stride 400 series machines which use the M68000 processor at 10 Mhz. If the 1 Meg by 1 chips replace the 256K by 1 chips, the one-board computer will have 8 Megabytes of on-board memory with no wait states.

The limiting factor in chip design is the spacing within the chip mask and the power available for driving the internal circuits. The power factor is limited by the heat dissipation within the chip. For a constant power factor within the chip, the speed of the circuits is a function of the mask spacing within the chip.

Six years ago, the 4-Mhz Z80 chip made a significant impact on the 8-bit processor market when the clock frequency standard at that time was 1 to 2 Mhz. Today, the market is preparing for the full 32-bit internal and external architecture, such as the MC68020, Intel's 80386, NC32032 and Z80032. Major advances in chip masking (down to 2 microns for the MC68020) made possible both the increase in chip complexity and the increase in processor clock frequencies.

With DOD asking qualified vendors to look into the 1.25 micron technology, a main-frame desktop calculator running at clock frequency of 40 Mhz with 20 megabytes of memory is a distinct possibility by 1995 if a commercial market is sufficiently strong in that area.

The traditional technologies of chip designers are N-channel metal-oxide semiconductor (NMOS) and KMOS, because of their high performance level. These devices, however, are power-hungry and require significant amounts of cooling.

Complementary metal-oxide semiconductor (CMOS) technology requiring less than one-tenth the power of NMOS offers a good solution to the chip heat problems. The drawback of CMOS is that it permits fewer devices within the die area, thus requiring more space per board. However, chip density on the board is greater for CMOS than for NMOS when cooling between chips establishes chip spacing on the board. A CMOS board for most applications does not require cooling fans.

As the computer complexity increases, so do the integrated circuits required to support microprocessors, disc controllers, and memory controllers. Custom gate arrays eliminate 40 to 50 integrated circuits in typical applications. The custom gate arrays have clock frequencies up to 25 Mhz.

Market pressures make strange bedfellows in chip technology applications. Emitter-coupled-logic (ECL) based integrated circuits almost disappeared from consideration by system integrators three years ago. ECL generated too much heat and was too expensive for all but mainframes or small specialized devices. Now with the push from microcomputers, the traditional Schottky TTL logic which was the basis for the VAX-11/780 has given way to the ECL technology, even in the face of the rapidly developing CMOS and gallium arsenide technologies. Both the Prime 9955 and VAX 8600 are ECL-based machines.

Another factor in the shrinking chip is a new semiconductor technology called wafer-scale integration (WSI) from Trilogy Systems. While a typical mainframe now uses 3500 to 4000 chips, the WSI machine would have just 40 wafers -- an order of magnitude reduction in chip count which is difficult to comprehend.

#### Peripherals

The explosive developments in computer peripherals are two to three times greater than the central processor developments. While a computer system contains one central processor unit, peripherals number five to ten items. A computer system may include one floppy (flexible disc) drive, one hard-disc drive, two printers, one plotter, one modem, one terminal or one monitor, and one keyboard. The peripheral add-on and update market is also significant.

Nothing exemplifies changes in the peripheral market more than hard-disc drives. Two years ago, a 10-megabyte, 5 1/4-inch, full-height (3.2 inches) Winchester disc drive sold for more than \$2000.00. This drive's average access time was in excess of 70 msec. Priam Corp. recently announced a 70-Megabyte (unformatted), 5 1/4-inch, half-height Winchester disc drive with an average access time of 25 msec. To equal this performance two years ago would have required an 8-inch, hard-disc drive at many times the cost of the Priam model 201 which is expected to sell for less than \$2000.00.

#### Software

Software capabilities and complexities closely followed the hardware and operating system capabilities, and the level of programmer access to the machine capabilities. Twenty years ago, the Univac 1107 and the IBM 7094 computers dominated engineering problem-solving activity. Disc drives did not exist. Drum storage devices were at a premium and were usually reserved for the operating system. The computing jobs were I/O and cpu bound.

The engineering/programmer submitted card input data. With the computer, card reader, and printer behind closed doors, access to rapid turnaround computing was limited to high-priority situations. When access was granted, the engineer quickly learned to operate the card reader and to load tapes on the tape drives. Under special circumstances, a printer was available as the job was running and the user could monitor the job progress by leaning over the printer. If the user could think in this noisy environment, a new job could be entered into the computer before the previous job was completed.

Even though access to the machine was limited, jobs submitted to the computer broke new ground in matrix operations, finite element formulations, redundant analysis, and time simulations. Many of the tasks which were previously the forte of the analog devices rapidly migrated to the digital computer. As software developers asked for more powerful machines, a general lack of understanding of software maintenance and structured programming techniques created software updating and portability problems. Since computing speed was a dominant factor in programming, the programmers avoided calls to subroutines as a matter of policy. Programs were primarily one large main segment.

Fifteen years ago, Winchester drive technology made possible disc drives with access times short enough to replace magnetic drums. This opened up practical use of program overlays and expanded the range of software applications that could be contained within one program. While memory address space was limited, codes which were an order of magnitude more complex than what was possible before overlays could be implemented on the computer.

Ten years ago, interactive computing on the mainframe made possible an explosion of software development and interactive analyses. Then on the opposite end of the computing power, minicomputers arrived on the market and provided engineering with interactive and batch computing at a cost many times less than the mainframes. A few years later, the advent of the superminicomputers opened up a floodgate of interactive applications.

The question of coding the algorithm gave way to the reality that the code had to be maintained. Structured programming entered, and more readable code replaced subroutine calls overhead. Software maintenance costs generated numbers many times the original software development. The question of how to program the task was replaced by the question of how to maintain the program and retain configuration control.

Today, software has taken over the cost side of the computing equation in some applications. More frequently, computing machines are selected because they support specific software programs. This shift from the data processing mentality to productivity awareness is directly attributable to specialized software such as circuit board design packages and engineering workstations.

The effective use of some software may require substantial training and certain skill levels. This is also a cost. In areas with high employee turnover, software which is user-oriented and offers an "expert system" approach could be a more cost-effective solution. The higher software one-time entry cost would offset the reoccurring higher training costs. Software today can store and disseminate the organization's expertise in a more usable form than the traditional technical memos of a few years ago.

#### COMPUTING TRENDS

The three primary functions in computing are: 1) database management, 2) processing, and 3) user communication. Five to ten years ago, these three functions were commonly found on one computer.

Structural design problems require access to large databases. The proper management and direct access of these databases are essential to cost-effective computing over the long term. Sufficient processing power for both interactive and batch function is also critical to cost-effective solutions; however, if engineering uses interactive terminals on the mainframe for its interactive functions, computing power for cpu or I/O intensive batch jobs will be limited. Therefore, large computing problems can be processed only at nonprime times. This is satisfactory if there is sufficient capacity during nonprime times, and a 12-hour turn-around will satisfy schedule requirements.

Interactive computing has a built-in growth factor. The more the user had access to interactive terminals, the more the user saw new applications. Soon, the mainframes became saturated with interactive functions during the prime time. Interactive response times increased, and batch processing throughput reduced to a trickle. A mainframe which was capable of performing large data transactions and significant numerical processing was not cost effective servicing interrupts from interactive terminals.

The solution to this problem appeared under two names:

- 1) distributive computing
- 2) front-end computers

If the user completes the task on the interactive computer, the facility is distributive computing. If the user prepares a job for the mainframe, the facility looks like a front-end computer. The more familiar of the front-end/distributive computers are the VAX 11/1780 and the IBM 4300.

Applications for the new-found computer access grew again as availability and user friendliness increased. Graphics applications in the form of postprocessors and highly interactive preprocessors were possible; however, as the applications grew, the interactive response times deteriorated. Consequently these applications were not fully implemented on the super minicomputers.

As direct access to the mainframes is removed, the mechanics of getting a job to run on the mainframe becomes more and more important. In many computer installations, job preprocessors on the front-end computer prepare a job stream for the mainframe. The job stream includes job control language (JCL), input data generators, and control data for data management. Even with good job preprocessors, however, the number of jobs to be submitted to the mainframe soon becomes the bottleneck in the design process.

Computer networks are a significant factor in superminis and workstations computing environments. The superminis manufacturers such as DEC provide their own network, VAXNET. Some workstation manufacturers such as APOLLO provide their own network, DOMAIN. Networks based on general LANs (local area network) are coming on-line. However, the hook-up price of \$2000 per microprocessor is still high when compared to the microprocessor cost of \$3500. If the market were to produce a full 32-bit processor with floating point co-processor and 4 Megabytes of memory for under \$15,000, then LANs would be cost effective. Each microcomputer could be used effectively during off hours by a master controller. Until now microprocessors have had limited applications in significant engineering problems.

## DISCUSSION

Computer technology, computer marketing, and management appear to be the primary forces that will determine the direction of the engineering computing environment during the next ten years. Software will act primarily as a constraint on hardware integrators and manufacturers. Since the cost of replacing or recoding existing software can exceed many times the cost of a new machine, new hardware offerings will be compatible at the computer operating system level with an existing machine.

Software developments during the next ten years will focus on improving the performance of current software by integrating into the engineering design methodology advance tools such as "expert" systems, advanced database management systems, and fifth-generation screen generators, where increases in productivity will offset the increase use of computer resources.

In standard structural and dynamic analyses, FORTRAN will remain the primary programming language. FORTRAN 77 will eventually replace FORTRAN IV and its variants, and will be upgraded to include some of the UNIX concepts such as trees and unions.

### The computer technology future

The computer chip revolution will continue, and it will affect the complete line of computer products, from microcomputers to supercomputers. The workstation computing power will increase, and special-purpose computer applications, such as spectral analyzers and sensor data processors, will explode.

The computing power of mainframes and superminis will increase, along with performance-to-price ratios for these machines. Hardware footprints will decrease. If the cost reductions of CMOS technology continue at their current rate, the superminis, if not the mainframes, will operate without significant cooling requirements.

Engineering workstations with increased computer power will process tasks that are currently running on superminis. These tasks will be interactive and graphic-display intensive.

The current superminis will be replaced by super microcomputer systems, probably running under some variant of UNIX. These will drive the engineering terminals for basic data entry and will probably front-end the mainframe minis and mainframes.

The power of the mainframes will increase to the point where interactive terminal applications may again be feasible. For this to be possible, the multiuser systems designed for interactive processing will be running under the native operating system, such as UNIX under IBM's VM system.

### Computer marketing future

Business applications will continue to dominate the computer hardware and software market share. The market segment for Crays and other supercomputers will remain essentially constant. Supercomputers require a constant flow of floating point intensive, vectorized codes in jobs where disk I/O is a minimum. While some engineering computing jobs fulfill this requirement, most do not. Therefore, ten years from now, the engineering environment will not change basically for problems requiring large databases and significant amounts of floating point calculations. The bulk of engineering design problems will remain on business machines.

While future high-tech design requirements may be the motivation for increasing the level of computer capacity, the computer technology, marketing, and management forces will establish the level of engineering high-tech which a design can economically support. High-tech engineering requires a high-tech computing environment. Engineering design problems are not solved cost effectively in earth orbit or after 200 high-tech vehicles come off the assembly line.

### Management

Management sees three principal issues for computer allocations: cost, cost, and cost. The unstable element in the word cost is that it implies true cost, while in fact, it means perceived cost. This perception is a function of accounting procedures, management experience, and relationship to the organization goals. A growth organization perceives cost relative to maintaining a specific rate of growth, while for an organization being liquidated, management perceives cost relative to maintaining maximum short-term profits.

This perception issue often dominates talks with customers on computer charging. A customer should focus on the organization productivity factor relative to the cost issue and not on one element of identifiable cost. Cost reductions should be real. The only cost is that of doing business. A close look at the computer-charging issue may find that the expense of tracking computer costs may exceed the paper savings the customer extracts from the contractor. In the end, the customer pays, or the organization fails.

The interaction between cost, organization goals, and management is the critical issue in establishing the productivity criteria for computer resource allocations. While each organization has its own process for the interaction, a number of common issues are obvious:

- 1) Management wants flexibility/elasticity in computer facility architecture. The optimum is instant expansion or instant contraction around a core of committed resources. The expansion or contraction should be transparent to the user community.
- 2) The expansion or contraction of the computer resources should be the basis of reducing project costs, meeting committed schedules, and satisfying design goals. The computer expansion requirements should be known at the time management commits to a project for a specific cost and schedule. Contraction of computer resources should be on an over-capacity or unused-resources basis, unless the organization intends not to support a specific technology base.
- 3) The expense of new hardware and updating existing hardware must reflect the cost of business and not an abstract increase in computer facility budget.

#### A blueprint for the future

The blueprint should accommodate the future and still direct the overall effort to improve the engineering computing environment, with special emphasis on software use. The challenge is to suggest a clear alternative which will be cost-effective and remain compatible with the established and known forces of computer marketing and management resource allocation practices.

The goal of the proposed blueprint, then, is to provide an approach which will:

1. lower engineering computing costs.
2. adjust to the cyclic nature of engineering computing resource requirements.
3. be compatible with the primary computing marketing force, business applications.
4. address the I/O-bound computing tasks as well as the floating point cpu-bound computing tasks with and without vectorized code.
5. give management greater cost control and flexibility while integrating new computer technology.
6. permit enlightened management to take advantage of the computer as a productive tool.
7. lower the risk of vendors trying to service engineering requirements.
8. promote software development which is modular, transportable, low-maintenance, and computer installation sized independent.
9. be compatible with existing software.
10. permit management to appraise software as a company asset.
11. increase the interactive computing capability.
12. integrate advanced developments in hardware and software technology, such as parallel processing.

#### AN APPROACH

The blueprint for improving the engineering computing environment during the next ten years must satisfy management and user goals and recognize simultaneously the realities of the computer marketplace (business) and computer technology advancements (controlled chaos). The issues for management are cost, flexibility, and control. The issues for engineering are good computer interfaces, adequate software tools, and access to the hardware resources required to satisfy project goals and schedules.

The ideal batch processing hardware should have the data record transaction efficiency and the operating system flexibility of an IBM 30XX, the floating-point scalar processing efficiency of a CDC 7600, the vectorized floating-point efficiency of a Cray, and the modularity of peripheral additions. Business and manufacturing are the primary customers of a typical computer installation. Therefore, the core computer function must be a highly efficient data-transaction machine.

The goals listed above do not necessarily imply that they are either sufficient or necessary conditions for a satisfactory computing environment. They are the starting point to the definition of a working blueprint.

Lacking an ideal batch processor, the issue is whether a basic business computer can form the core of a good engineering computing system which will address the resource requirements of the structural design during high and low levels of design activity. The proposed computing system uses the high-transaction (host) mainframe computer as the primary organizer of work to be done, of data to be managed, of output to be sent, of resources to be allocated, and of computing requests to be serviced.

In this configuration, the host mainframe is encircled by independent floating-point processing computers which have their own memory and disc units. The host mainframe while executing a job would identify a cpu-intensive, floating-point processing task,



package the cpu-intensive task, and send it to an auxiliary computer to be processed. The auxiliary computer could be a CDC machine, an array processor, a 1-to-5 MIPS 32-bit MC68020-based machine, or a Cray.

The key to this architecture is the modularity of the auxiliary computers and the ability of the host mainframe to function independent of the auxiliary computer. At low levels of activity, the auxiliary computers could be removed. Maximum effectiveness of new low-end computing chips could be used to provide more MIPS at lower cost. This architecture would also enable the cost-effective tuning of some of the auxiliary computers to perform functions such as matrix inversion and matrix multiply.

A blueprint for a computing environment to satisfy the objectives stated above depends on two critical components:

- 1) the definition of a floating-point computing engine (FPCE) interface with a transaction-based computer (host mainframe).
- 2) the definition of a computer processing system which will integrate:
  - a) the floating-point computing engines resources
  - b) the mainframe resources
  - c) existing independent computing software into a complete computing system without changing the existing software.

FPCE interface- The concept of a floating-point computing engine is not new. The Crays, Floating Point Systems array processors, and Ridge computers are in fact floating-point engines. They specialize in performing floating-point calculations at the super-computer, attached processor, and microprocessor levels. Some of these engines are tightly coupled with the host processor, and others are stand-alone computers which need a front-end processor to prepare the job. The issue here is the communication between the engine and the host mainframe computer.

If an interface were to exist, the computer hardware integrator could develop a computer which would address the floating-point computing environment, whether on scalar or vectorized code. If the standard FPCE interface were supported by the FPCE manufacturer, its product could be a potential add-on to any high-transaction machine that also supported the FPCE interface.

The economic and flexibility advantages of this system are obvious. During slow design activity, the computing organization would support engineering with a minimal number of these computing engines. As the design activity increased, the computing organization would attach more computing engines to satisfy engineering computing needs. The method of processing the job could be completely transparent to the user, or the user could route the job to go to one or more of the computing engines.

To be an effective interface, the host computer must see the interface as a cross between another device and a remote job entry port. The interface must have data transfer rates close to disc devices. If the computing engine is a single task processor, the full resources of the engine could be available to transfer data at the host interface speed. However, since the elapsed time between sending the job to the engine and receiving the results could be hours, some type of communication protocol which defines the status of the computing engines and the host computer would be required.

Figure 2 shows a typical host mainframe installation with an FPCE interface to an FPCE computer. The figure also shows the mainframe with a front-end super minicomputer and direct terminal access ports.

The FPCE interface is not defined at this time. It may be based on local-area networks (LANs) or it may be based on something altogether different. The functional aspects of FPCE are shown in Figure 3, where the interface commands to the engine define the source code, linked code to be loaded directly, data, time estimate, device allocations and the commands to inquire the host status, time estimate to completion, compiled and linked codes, and results. Special consideration must be paid to the problem if either or both the host computer and auxiliary computers should fail during the processing of the job submitted to the host.

Extended Batch User System- Software exits in primary two forms:

- 1) modular (executable) software as found in the UNIX-like operating systems where the input/output streams are well-defined and simple.
- 2) integrated software built around an internal (sometimes accessible from the outside) database as found in NASTRAN, where input/output streams are complex and difficult to assemble.

Figure 4 shows the contrast between modular and integrated software. Modular software stands by itself. As long as the input data is on the designated units and the output data can go to other designated units, the modular software computes. Integrated software has its own database. Data required to run routine "B" in Figure 4 may have to go through routine "A" where it is catalogued and put into a standard form which will be recognized by "B". The MAIN must read additional control cards to execute "B" and output the data to the outside world.

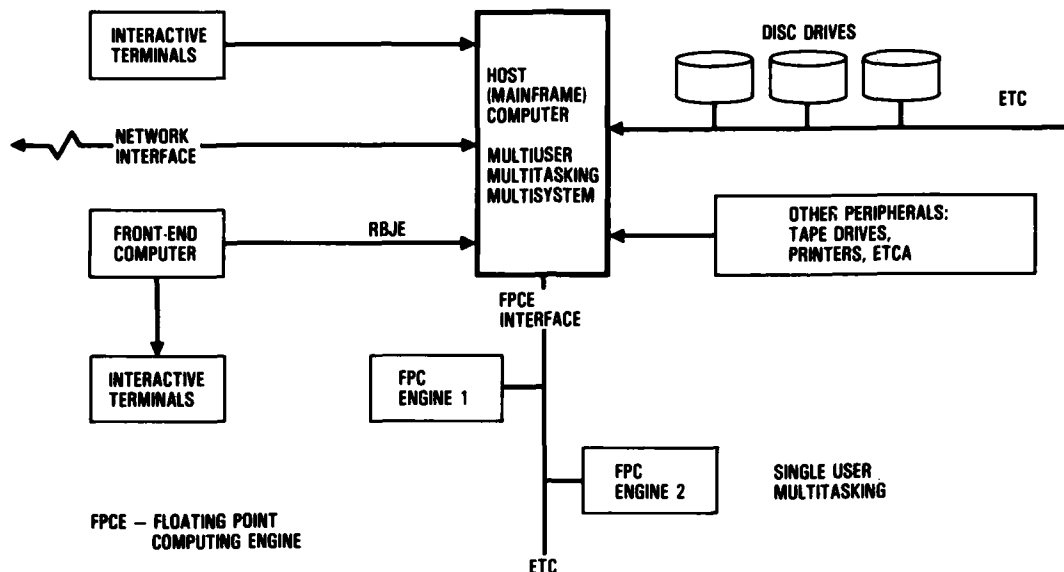


FIGURE 2. HOST COMPUTER WITH FPCE INTERFACE

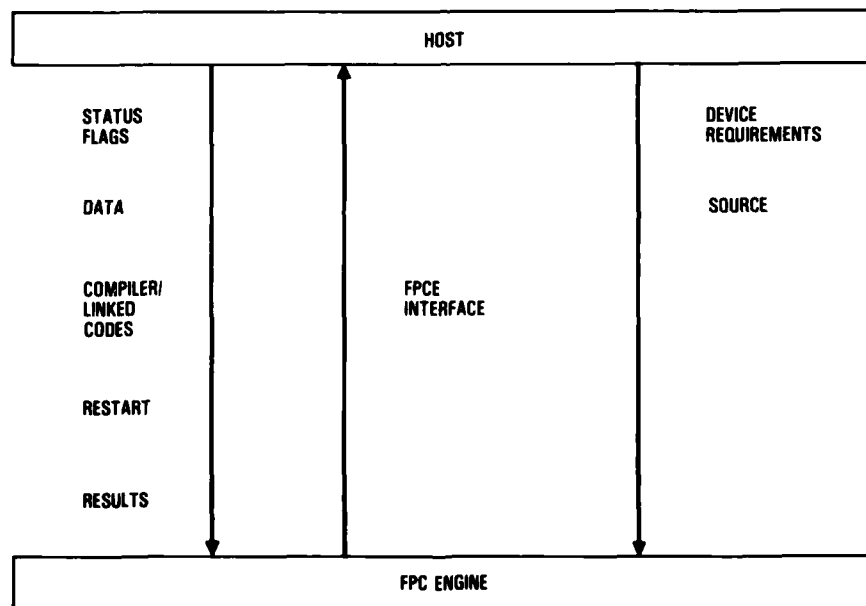


FIGURE 3. A TYPICAL FPCE INTERFACE DEFINITION

In the past, the integrated software approach was necessary to achieve efficient control over the execution of "A", "B", etc., routines. The integrated software provided either a self-contained attaching monitor system, as found in NASTRAN under the IBM 370 operating system, or OVERLAYS which contain SUBROUTINES called from the MAIN segment. The OVERLAY approach is used by many integrated software systems, including Boeing's ATLAS, Lockheed's FAMAS, and Lockheed's NICE.

The drawback of these systems is the very reason that they are so powerful, namely, complexity. It is not a trivial exercise to either add a function to these systems or to modify an existing routine. One of the NICE system features is that it is probably the easiest of all the integrated software systems to change or to add routines to. None of these integrated systems suggests the practicality of integrating one integrated software system into another integrated software system.

This was the problem that was addressed in developing an integrated software operating system called Continuous Batch User System (CBUS) at Lockheed-California Company. CBUS attaches to any executable load module and initiates execution. These modules include NASTRAN, FAMAS, and simple standalone programs.

CBUS permits the modular coding of software into executable load modules, which may be as small as two FORTRAN statements. These modules are coded and checked out on front-end computers where the interactive capabilities facilitate coding and debugging. These modules are then moved to the mainframe and integrated under CBUS with no modification to the source code. CBUS permits software development by many programmers without undue

concern about data interfaces. The macros in CBUS permit the definition of engineering processes which represent particular developments of specialists. An alter capability permits the user to define keywords which will trigger almost an unlimited number of changes to the macro with embedded defaults. The feature permits addition of functions to the process without changing the original definition.

CBUS uses the modules in the same form as that required by simple batch submittals. CBUS is an integrated operating system, which for the first time not only permits but encourages a computing function to be in modular format by its loosely coupled architecture. In its simple form, CBUS executes "A", "B", etc., as shown in Figure 5, in a continuous sequence. The modules "A", etc., represent any modules accessible to the computer.

CBUS is, however, more than an attaching monitor. It is a resource allocator and a user command processor; it executes user commands which represent a process or task. A simple one-line command may represent a complex process which requires the execution of a hundred equivalent batch programs. Each process is completely defined by the user in a MACRO structure. One set of commands under this operating system will execute the equivalent of 300 to 500 batch programs.

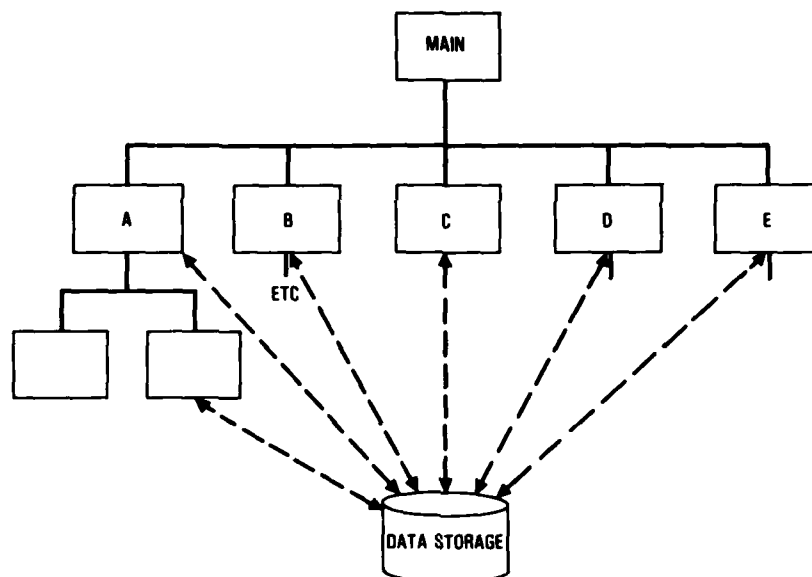


FIGURE 4. INTEGRATED SOFTWARE AND DATA MANAGEMENT FLOW

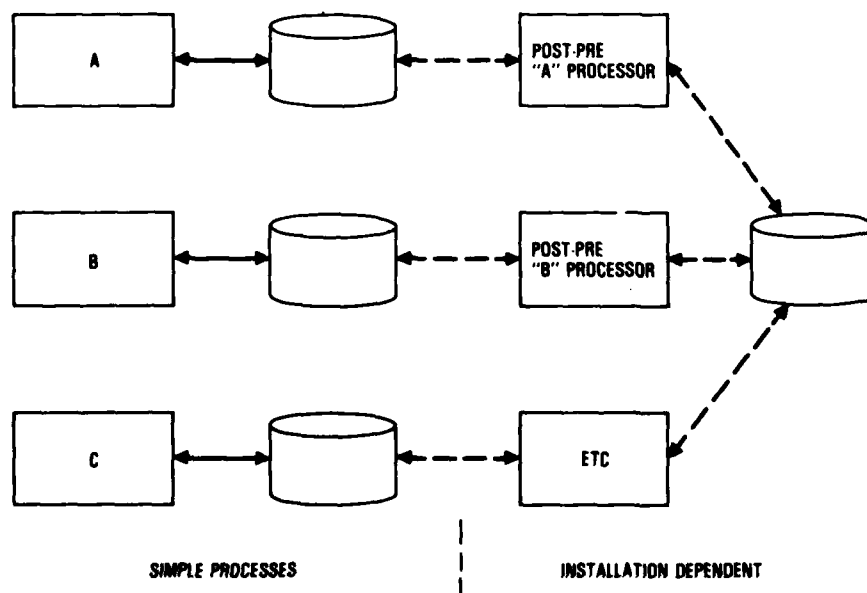


FIGURE 5. LOOSELY COUPLED PROCESSES UNDER CBUS

Figure 6 shows the basic elements of CBUS. Target programs are any executable load modules, which also include compilers and linkers. Within the command processor program (CPP), the alter capability is the key to the operational flexibility of CBUS. Alter capability permits the user to name a keyword which enables the MACRO to be altered during the execution of the engineering process. Some features of the command language are found in Figure 7 where simple commands are integrated into another MACRO called super-command. Figure 8 shows some of the elements of the alter capability.

The major benefit of this architecture is the control that user exercises over the problem being solved. The user not only defines processes which use complex integrated software such as NASTRAN, but with minimum effort, the user can insert a completely new process without interfering with anyone's use of the same macros. While CBUS is a structure to build integrated systems from a user point of view, it is not new; but has been used for the past seven years and has survived many operating system changes. References 1 and 2 discuss some of the CBUS details as well as its application to preliminary aeroelastic design of structures (PADS).

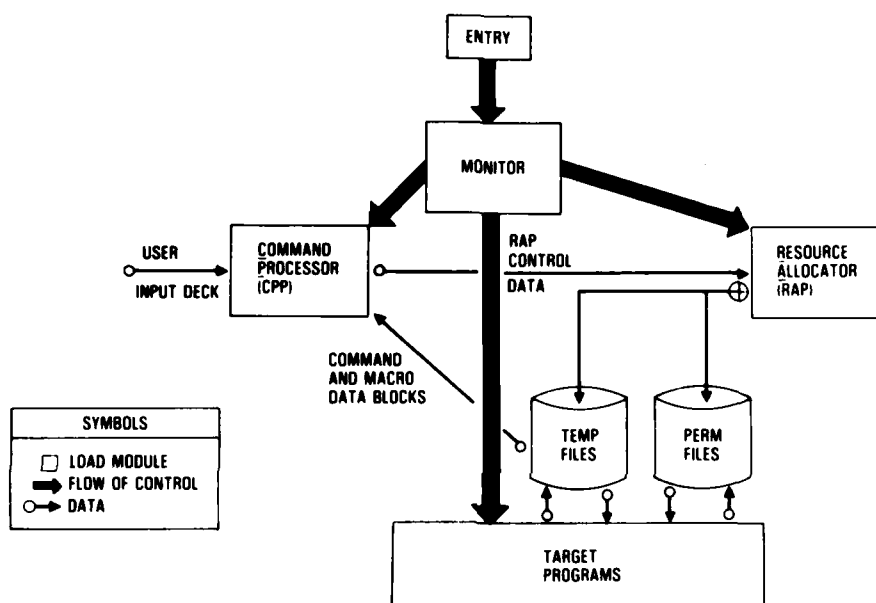


FIGURE 6. CBUS GENERAL ARCHITECTURE

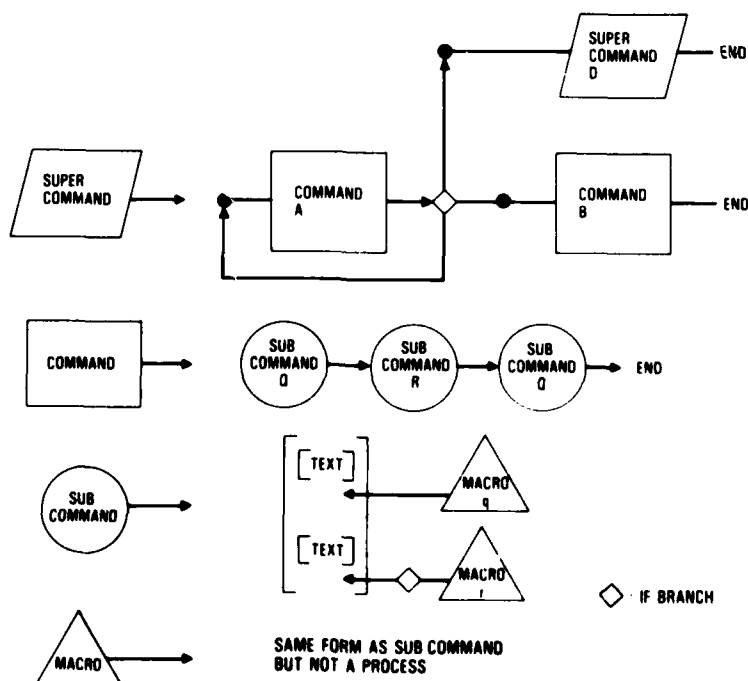


FIGURE 7. CBUS COMMAND/MACRO STRUCTURE

## ALTER CAPABILITY

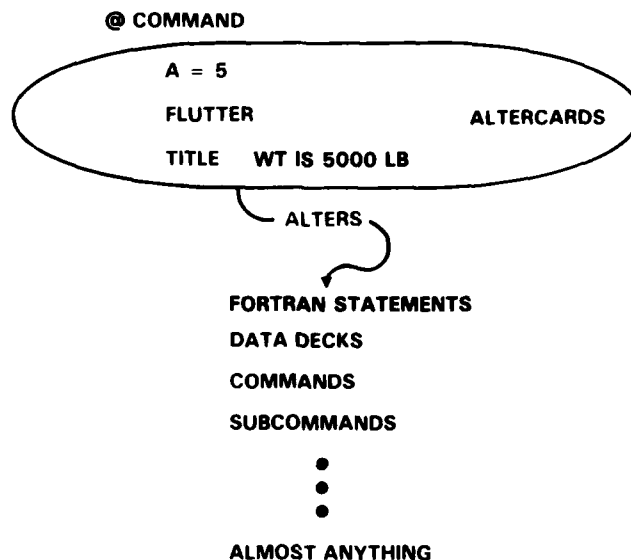


FIGURE 8. CBUS ALTER CAPABILITY

CBUS architecture needs to be extended for interaction with the FPCE interface. New features in the CBUS monitor, resource allocator, and command processor should include: 1) putting the computing task asleep when the mainframe must wait for the task on the floating point engine to be completed, 2) computing task wake up call when the engine task is completed, 3) Tees and unions to permit parallel processing on both the engines and mainframe. While CBUS is currently running only on the IBM 370 operating system, the software could be migrated to other mainframes. The concept is to build one integrating software operating system which will permit simple control and execution of modular software rather than to integrate the whole process into an existing integrated software such as NASTRAN. CBUS is an example of a software integrator which operates external to all programs running under its control.

The extended CBUS concept could be called XBUS.

### CONCLUSIONS

The computer technology developments are, and will remain for the foreseeable future, in a state of controlled chaos. Engineering cannot remain a passive observer of this process and expect the computer developments to automatically and cost-effectively address the needs of the engineering design process. Engineering must review its requirements and define a market segment which computer system integrators will recognize as an area with acceptable risks.

In software, engineering must recognize the need for more modular software architecture, which is not possible in a large integrated computing program. Engineering needs change and the software integration procedures must accept, as the basic premise, flexibility and change.

Engineering must control its computing environment in the face of serious economic pressures created by a rapidly changing computing industry which views the engineering computing needs as a small footnote in the computing marketing picture.

The following are observations and conclusions in support of the above statements:

- 1) The digital computer market is dominated by business applications.
- 2) Business machines are cost-efficient, data-transaction processors and database machines.
- 3) Business machines are not cost-efficient, cpu-intensive processors.
- 4) Future computer technology developments will continue the current trend of increasing performance-to-cost ratios.
- 5) Superminis processing power will increase to current mainframe levels within 10 years.
- 6) Microcomputers processing power will increase to current superminis levels within 2 years.

- 7) Structural design process requires the data-transaction power of a business machine, the scalar processing power of a CDC machine, and the vector processing power of a Cray.
- 8) Management perception of computing costs is a major factor in establishing the engineering computing environment.
- 9) Management desires computing capacity elasticity and computing cost effectiveness from a computer facility.
- 10) Software costs are in some applications the major element in the computing cost equation.
- 11) Current integrated software packages are difficult to maintain and modify, and are not portable.
- 12) A software integrator, called CBUS, permits the integration of diverse computer programs and integrated software systems such as NASTRAN without altering the code of the programs being integrated.
- 13) Cost-effective, floating-point computing engines (FPCE) which are based on complete 32-bit microprocessors are technically feasible.
- 14) An FPCE interface definition is required to stimulate the computer system integrators to supply engineering with the modular computing cells. The interface is between the host computer and the FPCE computers.
- 15) An extension of the CBUS system will permit the integration of the floating-point computing engine as another computing resource as well as disk, tape drives, and memory.
- 16) A computer facility based on a host computer (transaction processor) and FPCE computers would reduce the cost of a cpu intensive task by an order of magnitude. The host computer performance would not be seriously degraded while servicing engineering cpu-intensive tasks. The engineering jobs would continue to benefit from the cost-effective data management and transaction capabilities of the host computer. Increasing or decreasing cpu-intensive computing requirements from engineering could be satisfied by the addition or removal of FPCE computer units without affecting the operation of the host computer.
- 17) The CBUS concept permits the development of software in loosely coupled modular units which can be integrated into a single named process. The integration is at the user level and is controlled by the user. Inserting a new element into the process or changing an element of the process is trivial.

#### RECOMMENDATIONS

- 1) Form an engineering user group to establish requirements for hardware development which will make engineering computing more cost-effective.
- 2) Investigate the potential of forming an interface protocol between the host computer and auxiliary computers.
- 3) Investigate the potential of a software integrator system architecture which was used in CBUS.

#### REFERENCES

- 1 Radovcich, N. A., "Preliminary Aeroelastic Design of Structures (PADS) Methods Development and Application," Presented at the 56th meeting of the AGARD STRUCTURES and MATERIALS PANEL Specialists meeting; London, United Kingdom 10-15 April, 1983.
- 2 Radovcich, N. A., "Some Experience in Aircraft Aerolastic Design Using Preliminary Aeroelastic Design of Structures (PADS)," NASA Symposium on recent experience in multidisciplinary analysis and optimization, Hampton, VA., April 24-26, 1984.

THE NEEDS OF THE USAF AIR LOGISTICS CENTERS  
FOR LARGE SCALE STRUCTURAL ANALYSES AND  
SUPPORTING DATA

by  
Dr. Thomas F. Christian, Jr.  
System Program Management Division  
Warner Robins Air Logistics Center  
WR-ALC/MMSRD, Robins AFB GA 31098-5609 USA

## SUMMARY

This pilot paper reviews the structural analysis capabilities both now existing and projected as needed within five years at the USAF Air Logistics Center in order to support USAF operational aircraft. The various structural analysis and supporting data areas addressed in this paper as part of airframe force management include life prediction techniques, finite element modeling, electronic data bases, risk assessment and newly introduced advanced structural materials.

## INTRODUCTION

"I always avoid prophesying beforehand, because it is a much better policy to prophesy after the event has already taken place" - Winston Churchill.

At the risk of disregarding this sage advice of Mr Churchill, prophesy concerning the structural analysis requirements at the U.S. Air Force's Air Logistics Centers (ALCs) during the next five years does need to be made now. A path that can not be seen, even dimly, is a path that may lead to a tumble. As the repair depots and industrial facilities of the Air Force, the five ALCs are now ready to dramatically enhance their technological expertise in the structural analysis arena. This paper will focus on the various areas in which efforts are now being made such as life prediction, finite element modeling, risk assessment, flight history recording and introduction of new materials into greater operational use.

It must be remembered that until recently the structural analysis requirements at an ALC were quite uncomplicated. This was due to several factors. First and foremost, the Air Force Systems Command (AFSC) conducted new aircraft acquisition through the Aeronautical Systems Division (ASD) at Wright-Patterson AFB OH with research and development support from the nearby Flight Dynamics and Materials Laboratories. With the preponderance of structural engineers, it was and is the charter of AFSC to ensure an adequate airframe is delivered to the Air Force. Working in conjunction with the airframe manufacturer's research and development engineers and design and manufacturing staffs, AFSC has performed this function well. Once an aircraft enters operational service, it becomes the responsibility of the Air Logistics Centers of the Air Force Logistics Command (AFLC) until its retirement. As a result of this approach, for many years structural analysis at the ALC's consisted of referring to the stress analysis reports furnished by the airframe manufacturer and ensuring that any proposed repair would restore sufficient strength. Relatively simple calculations of P/A and Mc/I were often all that were necessary to satisfy the ALC mission of supporting in-service aircraft and keeping them airworthy. As aircraft have remained in operation longer than originally foreseen when they were designed, structural fatigue and corrosion have assumed greater prominence and required greater engineering attention at the ALCs. Added to this is the multiplicity of unanticipated missions which have been acquired by virtually every series of military aircraft and which have caused the need for more engineering resources.

Diversity and age of airframe configurations within an aircraft force, such as the C-130 transport which has been in production since 1955, also plays a role in requiring the enhancement of structural analysis capabilities at the ALCs. The current efforts and future needs in each of the structural analysis disciplines at the ALCs will now be delineated.

## LIFE PREDICTION FOR FATIGUE AND FRACTURE

"To repair the irreparable ravages of time" - Jean Racine.

For the foreseeable future, fatigue life prediction will occupy the predominant position among the structural analysis needs at the Air Logistics Centers. Each Air Logistics Center has several System Program Managers (SPM) to support the aircraft forces assigned to it, for instance, the Warner Robins Air Logistics Center (WR-ALC) is responsible for the C-130, C-141, F-15 and all Air Force helicopters. A principal SPM activity for ensuring the structural health of operational aircraft is known as the Aircraft Structural Integrity Program (ASIP) as codified in AFR 80-13.<sup>1</sup> The requisite functional tasks to ensure adequate airframe quality are delineated in MIL-STD-1530A<sup>2</sup> as design criteria, design analysis and development tests, full scale testing, force management data package and force management. The airframe manufacturer and ASD are responsible for the first four items and the last one is the purview of the ALC. These five tasks substantiate the structural integrity of the aircraft design, assess the in-service integrity of individual airplanes, determine logistics requirements and improve future design methods. Central to all of activities are the concepts of durability and damage tolerance assessment (DADTA)

established with the issuance of MIL-STD-1530A and its associated military specifications, MIL-A-834443, MIL-A-8866B4 and MIL-A-88678B5.

Damage tolerance is defined by the U.S. Air Force as the ability of an airframe to resist failure due to the presence of flaws, cracks, or other damage for a specified period of unrepaid usage. Damage tolerance analysis (DTA) thus becomes an investigation of crack growth and the residual strength possessed by the damaged structure. It focuses on primary or "safety of flight" structure which is assumed to contain initial undetected flaws which could likely be missed during an inspection. The safety limit at a given location on an airframe is then the number of flight hours required for a crack to grow from the assumed flaw size to a critical, or a catastrophic failure, crack length. Durability, on the other hand, is not a "safety of flight" concept but rather an economical one which measures the ability of the airframe to resist cracking, corrosion, wear, etc. for a specified period of time. Durability limits are associated with the cost of in-service repair such that initial flaws, which are not safety of flight, will not grow large enough to require extensive repair before one service lifetime.

Under present procedures, the airframe manufacturer conducts an initial baseline or total airframe DADTA under Air Force contract (see, for example, References 6 and 7). This may require 15 to 20 contractor and Air Force engineers from one to three years depending on the specific airframe complexity<sup>8</sup>. The number of critical locations varies but generally speaking larger aircraft have more than smaller aircraft, perhaps over 100 for a bomber/transport class and less than 100 for a fighter/attach type. Regardless of the number of locations considered in the baseline study, it is obvious that other, unanticipated locations will surface during operational service. The multiplicity of mission changes subsequent to a baseline study as well as the prohibitive cost of an inclusive study make such occurrences inevitable. For this reason, the ALCs must have the capability to analyze such locations organically.

Conducting a flight-by-flight crack growth analysis organically at an ALC requires of course the same procedures as undertaken by the airframe manufacturer in the baseline study. It requires incorporation of the following elements:

- (a) the initial flaw's size and shape
- (b) the aircraft usage stress spectra
- (c) data for material fracture toughness and constant amplitude crack growth rate
- (d) crack tip stress intensity factors
- (e) damage integration and load interaction models
- (f) the fracture criterion

These elements<sup>9</sup> are shown schematically in Figure 1. Naturally, the result of this analysis is the crack growth curve, as illustrated in Figure 2, from which the length of the crack at any time before the safety limit can be obtained. This then provides the information necessary to select non-destructive inspection (NDI) techniques and to set inspection intervals or to plan structural modifications.

The Air Force experience with DTA during the decade since its introduction<sup>10</sup> has been good. The ALCs are the beneficiaries of the outstanding efforts of the airframe manufacturers and the AFSC organizations which have pioneered this methodology as a replacement to the previous linear accumulation fatigue damage approach. As this methodology has been made available to the ALCs, concomitant decisions have been necessary as to the level of organic DTA desired at a given ALC. The choice then determines the requirements for large scale structural analysis. Five possible levels<sup>11</sup> have been identified as follows:

Level I: Review In-House. This is the most primitive level of organic DTA effort. It requires no structural analysis software or hardware capability since ALC engineers merely review work performed by contractors.

Level II: Structural Analysis of Repairs. This level introduces finite element analysis techniques on a minicomputer or time sharing on a larger mainframe. Graphics terminals employing a pre and post processor to generate structural models and display output results such as deformations and stresses are highly desirable.

Level III: DTA on Structural Components. Building on the hardware resources of Level II, this level adds flight-by-flight fatigue crack growth calculation capability using software such as the USAF Flight Dynamics Laboratory (AFFDL) developed CRACKS series<sup>12</sup>. With the modification or development of stress spectra and mission mixes, predictions of safety limits and establishments of inspection intervals becomes possible.

Level IV: Abbreviated Airframe DTA. This level requires specific flight-by-flight crack growth software for each aircraft type being analyzed. Preferably, this software will be developed by the airframe manufacturer as part of the initial DTA on that aircraft. A technical specialist for each aircraft type is highly advisable.

Level V: Complete Airframe DTA. This final level of organic DTA capability utilizes the finite element and flight-by-flight crack growth software of the previous levels for a



comprehensive DTA effort. It is essential for such an effort that both a number of highly experienced engineers and computer scientists as well as a dedicated computer facility with at least a super minicomputer be available.

At the present time the five ALCs vary considerably in their organic DTA capabilities but it does not appear farfetched to predict that within the next five years all the ALCs could be at the IV/V plateau which WR-ALC has achieved. This cannot be accomplished, however, without the close cooperation of each ALC with the various airframe manufacturers of its assigned aircraft and the cognizant AFSC organizations such as ASD and AFFDL. A strong partnership between these groups recognizes that an ALC's organic DTA capability will help achieve the common goal of airframe structural integrity. The enhanced proficiency of ALC personnel promotes a mutually beneficial exchange of expertise. This rapport must be fostered since DTA at an ALC cannot supplement the role of AFFDL to develop improved life prediction methodology. Nor can it supersede the need for close, continuous airframe manufacturers involvement over the lifetime of the aircraft due to the unique information and analysis resources possessed by the manufacturer. Using manufacturer developed aircraft specific software at the ALC to vary mission profiles and mixes, determine flight loads and stress spectra, and finally calculate crack growth curves cements a good working relationship between the engineering staffs as well as also minimizing the possibility of divergent results for similar analyses. In this vein, the experience of the organic DTA group at WR-ALC with its counterparts at the Lockheed-Georgia Company, the McDonnell Aircraft Company and the Sikorsky Aircraft Company has been very satisfactory. Equally positive has been the ability of WR-ALC to apply DTA methodology to the practical requirements of day to day force management and to maintain a rapid response capability for operating commands' needs.

Specific needs for organic DTA within the next five years focus upon user-friendly, menu driven integration of the elements shown in Figure 1. An excellent first step is the CRKGRO program<sup>13</sup> developed by Rockwell under AFFDL contract. It has a very good selection of stress intensity solutions for various crack shapes in flat plates. In practice, it is the determination of stress intensities for the various airframe details that is the most tedious and time consuming portion of the DTA analysis. Present techniques<sup>14</sup> require either compounding classical solutions such as back surface modification, finite width effect, etc. or the construction of a detailed finite element model to use a crack tip element, strain energy release rate, crack opening displacement, etc. Consequently, further work is needed to develop a stress intensity solution or "Beta factor" data base that can be used to accurately and rapidly establish geometric correction factors for configurations more complex than a plate with or without a hole.

Work is well underway to build extensive material property data bases for integration into the crack growth analysis software. Basically, the most frequently used aircraft structural alloys such as the 2000 and 7000 series aluminums, titanium alloys and some steels have been fully characterized. Of course, several lesser used airframe materials such as magnesium, many steels and a few aluminum alloys still require additional testing to quantify crack growth and fracture behavior. Mention should also be made of the classical material fatigue or crack initiation computer programs<sup>15,16</sup> now developed. Coupled with crack growth analysis, crack initiation serves to indicate areas of concern and to conduct rapid parametric studies of the effects of material and stress level changes. Such studies are now becoming more important at the ALCs to evaluate the relative benefits of potential repairs. Thus usage of these studies should increase during the next five years as aids in quickly formulating force management decisions.

#### FINITE ELEMENT MODELING

"Man is a tool-using animal . . .

Without tools he is nothing, with tools he is all" - Thomas Carlyle

Probably the most versatile tool presently in the stress analyst's repertoire is finite element modeling (FEM). As was the case with DTA, FEM was utilized by aircraft manufacturers, ASD and AFFDL engineers well before its use spread to the ALCs. Now the role of FEM at the ALCs differs only in degree from its usage at these other activities. By the very nature of their responsibilities, the ALCs must react swiftly to structural questions. Hence, a finite element model must necessarily be kept to a size commensurate with the time constraint of the given situation. With that caveat, FEM can cover a gamut of uses at the ALCs. The primary purpose is, of course, to adequately describe the stresses in a particular structure either to determine the effect of repairs and modifications or to calculate crack growth.

The partnership between the ALCs and the aircraft manufacturers is especially strong in the realm of large finite element models. Generally speaking, the airframe contractor has a much larger engineering staff and far more extensive computing facilities. For this reason, large finite element models, say over 5,000 degrees of freedom, are usually done under contract. This is obviously true of the entire airframe coarse finite element models, such as Figure 3, which are constructed in order to obtain internal loads for the baseline DTA studies. The predominantly used approach is to apply unit loads to the model and evaluate the resulting fractional stresses produced in each element. In this manner, stress-to-load ratios or influence functions are obtained which directly relate external airframe loads to internal structural stresses and displacements. Once these very large coarse models are transferred to the ALC they can then be used to provide the forces and displacements required as boundary conditions for finer detailed models created at the ALC to study a given problem.

The smaller detailed models generated at an ALC satisfy several needs. The determination of stress intensity solutions via FEM has been alluded to earlier in this paper. Basically, structural features with complicated geometry and load transfer lend themselves to models employing crack tip elements to derive the "Beta factor" for the crack growth analysis. Such models also quickly establish critical crack lengths. Even without the use of singularity elements, detailed models are useful for investigating the local stress reduction/redistribution due to a repair, modification or failed member. These stress results can then be applied, either as changes to the far field stress spectra or as stress to load ratios, in a conventional flight-by-flight crack growth analysis. Even when cracking has not been observed in-service at a suspected location, detailed modeling at an ALC can aid in deciding on the potential crack paths by mapping the tensile stress trajectories. This is often necessary to locate "hot spots" and to suggest possible NDI techniques. Furthermore, these models can also demonstrate if sufficient load carrying capability still remains and if flight restrictions will be effective in alleviating the situation. Corrosion grindout limits are also established more accurately by using FEM. Finally, the use of FEM at the ALCs is a valuable adjunct to flight testing. It is often difficult to locate accelerometers or strain gages at the precise position desired for a stress analysis due to the physical confinements of the actual structure. The finite element model provides the interpolation map which makes possible minimizing expensive flight tests.

At the present time, FEM at the ALCs appears to be focusing on the use of NASTRAN, primarily the MACNEAL-SCHWENDLER Company (MSC) version, as the main finite element code. This is in keeping with the entire aerospace industry's current direction. Although many major mainframe manufacturers developed their own programs a decade ago or highly modified the early versions of COSMIC NASTRAN, the recent trend has been to standardize on MSC/NASTRAN. From an ALC point of view, such standardization is quite beneficial since the manpower and computation limitations at an ALC preclude adroitly working simultaneously with several different finite element codes. The normal transfer of military personnel on a four year cycle, as well as the routine demands of other activities on both civilian and military engineers, means that organizational efficiency is best served by having only one large finite element code in use at an ALC. Of course, most large codes have dynamic, aeroelastic and buckling capabilities in addition to static analysis but the same reasons that tend to obviate doing more than one such large code also tend to obviate sloping these types of analyses routinely at an ALC. The more likely event would be to contract such work to the airframe manufacturer.

A similar focusing trend in computer hardware acquisition is also discernible. The tendency is toward use of dedicated supermini computers for structural analysis with several ALCs having or obtaining the DEC VAX 11/780. As was pointed out in reference 17, such machines provide user-friendly operational characteristics such as interactive operating systems, local control over resources and turnaround, and high-speed graphics but at the price of being considerably slower than mainframes. For example, one contractor developed entire airframe MSC/NASTRAN model which took 30 minutes to run on the contractor's mainframe takes nine hours to run alone on the WR-ALC VAX 11/780. Of course, such runs are not daily occurrences and the use of a model with superelement formulation can improve such run times significantly. Nevertheless, the benefits of a dedicated super-minicomputer to an ALC engineering and reliability branch now appear to well outweigh the disadvantages. One point that must be made, however, is that regardless of the machine chosen by an ALC there will always be a significant effort in software conversion for programs written for other machines. In spite of the fact that Fortran is almost universally used for scientific programs, there will always be machine unique coding that causes time consuming translation. The ALCs are, as was stated earlier, the beneficiaries of the considerable efforts undertaken by ASD, AFFDL and the various manufacturers but much of that work's software still had to be converted before it could be used at an ALC.

In contrast to the finite element code and super-minicomputer utilization, the pre and post processor graphics hardware and software arena shows great diversity. The key to the rapid construction and the prompt evaluation of the results of detailed finite element models at an ALC lies in this area. Several of the major airframe manufacturers are developing pre and post processor for FEM in conjunction with their computer aided design/computer aided manufacturing (CAD/CAM) efforts. Although these efforts hold great future promise, they have not yet reached the point of being available for use at the ALCs. In most instances, they are written for use interactively with a large mainframe or for use with the more expensive and large "smart" terminals. Presently, the ALCs are investigating the commercially available pre and post processor graphics packages, primarily PATRAN-G and SUPERTAB. In regard to hardware, a myriad of color graphics terminals are now on the market. They cover a wide spectrum of features, capabilities and price and each ALC will have to decide which best serves its needs. At WR-ALC, SUPERTAB and TEKTRONIX 4105 color terminals were selected but there were many other equally good choices. Currently, there does not seem to be a trend toward commonality among the ALCs in this area.

The needs of the ALCs in FEM during the next five years appear somewhat utopian when stated in black and white. The greatest FEM deficiency which now faces an ALC is the laborious and slow process of converting blue print dimensions to a model. There does not appear to be any technology on the horizon to replace this manual effort. The employment of technicians or engineering aides, such as undergraduate engineering students in work/study (co-op) programs, can relieve more experienced graduate engineers but it still does not condense the time required. The problem, or challenge if you will, is that all of the aircraft now operational were designed before the current computer aided

engineering (CAE) capabilities were developed. Added to this, data storage methods such as microfilm make it virtually impossible to use line tracing techniques due to the unavoidable distortions in replicating the original scale drawings. Consequently, the tedious work of "reading" the drawings and "building" the model with its concomitant time delays appears inescapable. The next aircraft designed and built using CAE will have electronic data bases for dimensions which will make FEM both quick and simple. Unfortunately, it is unlikely that there will ever be sufficient time and money to create such electronic data bases for older aircraft.

#### ELECTRONIC DATA BASES

"Obviously, a man's judgment cannot be better than the information on which he has based it" - Arthur Hays Sulzberger

The introduction of electronic data bases for aircraft structural information at the ALCs has the potential to markedly improve force management procedures and posture. Airframe force management is defined<sup>18</sup> as the specification and direction of inspections, preventive maintenance, repairs, modifications, and damage assessments required to economically prevent structural failure and preserve the strength and rigidity of the individual airframe during its useful life. Obviously a crucial electronic data base for force management is one containing crack length versus flight time curves calculated by DTA for the various airframe critical locations. Since these curves were calculated using force averages for mission stress spectra and mission mix, as shown in Figure 1, then it is possible to adjust these curves based on the stresses experienced by each individual aircraft. The process which accounts for individual aircraft flight loads and operational environment is known as the Individual Aircraft Tracking (IAT) Program. By using flight logs, counting accelerometers and strain recorders; the specific usage of each aircraft can be determined, and consequently, so can be found the predicted crack growth due to that usage. Computerized IAT data bases of this type<sup>19</sup> now exist based on crack growth and older software based on fatigue initiation are being converted to crack growth.<sup>20</sup>

The next step to be taken by the ALCs in using the computerized IAT electronic data bases is to modify the computer programs so that the system is capable of being updated by maintenance feedback<sup>21</sup>. In this way, future maintenance action projections can account for past inspection results, added repairs, and airframe modifications. Feedback from inspections is particularly important since the crack growth curves are predicated on an assumed initial flaw size. If an inspection does not reveal a crack at a particular location then analytically this means the location did not have an initial flaw as large as assumed. Hopefully, this will be the case for the vast majority of aircraft inspected and the crack growth curves for these locations can be reset to the NDI detectable flaw size. The result is a shifting of the crack growth curve, as shown in Figure 4 from reference 18, to a higher flight hour safety limit. Therefore, credit can now be taken for better manufacturing quality than could be assumed in the baseline DTA.

The IAT computerized data base should also be able to supply information on usage severity by home base or mission type. Such information on a continuing basis allows damage leveling among aircraft by switching high damage aircraft among bases or by having certain aircraft avoid high damage inducing operations. Additional updates to the force structural maintenance plan, supplied originally by the airframe manufacturers as part of the aircraft procurement, are also a potential result of the information being acquired by the IAT program. If the IAT data base reveals that unanticipated cracks are being found and reported during aircraft inspections or that significant changes from the baseline DTA stress spectra are occurring in aircraft operations, then the aircraft structural inspection schedules should be changed accordingly. Ultimately, the optimum airframe inspection schedule for a particular individual aircraft tail number can be created using the IAT data base. Of course, considerations of non-airframe inspections could modify such schedules so that advantage can be taken of aircraft downtime for engine overhaul or avionics repair.

In order to accomplish the tasks possible with the IAT data bases, two present needs must be fulfilled. First, the current tracking systems must be improved significantly to reduce time lags, bad data and excessive manpower requirements. Second, a comprehensive computerized system must be developed to identify individual aircraft inspection requirements. Such a system should be capable of accounting for variations in usage severity among individual aircraft as well as for peculiar repairs or structural configurations on any individual aircraft. Fortunately, work is now progressing on both of these needs. Flight loads data recorders are now in development<sup>22,23</sup> utilizing solid state microprocessor technology. Equipped with memory and data editing algorithms, these recorders can be "milked" or downloaded via portable ground interrogation devices for terminal transfer to the appropriate ALC. A computerized IAT system is now being implemented at WR-ALC with the C-141 transport as the prototype aircraft. Known as the Aircraft Information Retrieval System (AIRS), it will have approximately 10 years of data on each individual aircraft residing in its data base at any time. The data base can be accessed from special terminals, formatted in various modes and used for trending analysis with graphical displays. One such IAT data base operation is the Usage Simulation and Evaluation (USE) program which provides the capability to evaluate the effect of contemplated mission profiles and proposed utilizations on inspection requirements and damage rate. The inputs and outputs of these operations are shown in Figures 5 and 6 taken from reference 24.

## RISK ASSESSMENT

"Take calculated risks. That is quite different from being rash" - George S. Patton

The calculation of the risk of structural failure is becoming an item of keen interest and need at the ALCs. A quantified indication of the risk involved with a particular option should provide quite useful in the arduous decision making process that often occurs in force management. The work of Lincoln<sup>25</sup> computes the single flight probability of failure, the aircraft failure probability, and the expected number of failures in the force provided that specific information is available to the analyst. This information includes probability density functions for crack length and stress, critical crack length versus stress, inspection reliability, crack growth versus time and fastener hole reaming as the only rework decision. Generally speaking, however, all of this information is not available for a given situation. A somewhat similar approach recently broached by Saff and Forness<sup>26</sup> shows promise. Inspection results of failed and cracked components are normalized based on usage severity. The normalized lives of the cracked components are extrapolated to predicted failure lives using DTA and combined with the actual failure lives. The data for uncracked and unfailed components is accounted for as suspensions while a Weibull statistical technique predicts failure life distributions. From the Weibull distribution the risk of a structural failure can then be found for any point in the service life of a given aircraft or set of aircraft. Combining this technique with the DTA and IAT data bases under development could be a true boon to the SPMs at the ALCs. Much work remains to be done but this appears to be a most fruitful area for efforts during the next five years.

## NEW ADVANCED STRUCTURAL MATERIALS

"Nothing quite new is perfect" - Marcus Tullius Cicero

The advantages of advanced composite materials of higher strength-to-weight ratios, fatigue and corrosion resistance and aeroelastic tailoring potential are well recognized. Similarly, the metal matrix materials are also widely known to have benefits yet to be reaped. As these new advanced structural materials gain greater utilization in USAF aircraft primary structure, the ALCs will be called upon to repair such structure and to verify the repairs by analysis. It is beyond the scope of this paper to delve into the various failure modes or repair techniques under current study for advanced composites; there have been several recent conferences and meetings devoted just to these subjects as given in references 27 to 29. Supportability of composites is now being investigated within AFLC<sup>30,31</sup> and AFSC has contracted for studies of both the durability<sup>32</sup> and damage tolerance<sup>33</sup> of composites in primary structure. It is difficult at this juncture to predict the ALC's structural analysis needs for these new materials but it appears likely that they may be substantially greater than for current materials. Of course, much of the present methodology such as finite element modeling and flight loads spectrum development is readily transferrable but much fundamental work is still underway on such areas as impact damage and repairability at depot and field levels.

## CONCLUDING REMARKS

"Forewarned - forearmed" - Miguel De Cervantes

The next five years should bring great strides in structural analysis capabilities at the ALCs. Indeed, such advances are mandatory if force management is to keep pace with the requirement to maintain maximum operational readiness. The former Commander of AFLC, General James P. Mullins, has stated it in this fashion, "From the logistics perspective, the interesting thing about new technology is that it tends to feed on itself, generating greater support requirements which, in turn, demand even greater use of technology. State-of-the-art weapon systems today need state-of-the-art logistics support and lots of it".<sup>31</sup> Such state-of-the-art support can only come about by close cooperation between the ALCs, ASD, AFFDL, and the various airframe manufacturers. It is not difficult to project that within the next five years there will be computer data links between an ALC and its respective airframe companies to provide rapid transfer of flight spectra, crack growth evaluations and finite element analyses. Much needs to be done to reduce the mundane drudgery associated with many present analysis techniques and personnel at all levels within the affected organizations need to appreciate the new vistas that can be opened by their cooperation. Now that the Air Force has placed supportability on an equal level with performance and cost, the ALCs will have to assume a structural analysis role requiring a technical prowess heretofore unnecessary. There is every reason to believe that the challenge will be met. Perhaps the best way to conclude this paper is by suggesting that structural analysis at the ALCs is becoming the fulfillment of these words by Alexander Pope: "Be not the first by whom the new are tried, nor yet the last to lay the old aside".

## REFERENCES

1. Air Force Regulation, "Aircraft Structural Integrity Program", AFR 80-13, 16 July 1976.
2. Military Standard, "Airplane Structural Integrity Program, Airplane Requirements", MIL-STD-1530A (USAF), December 1975.
3. Military Specification, "Airplane Damage Tolerance Requirements", MIL-A-83444 (USAF), July 1974.
4. Military Specification, "Airplane Strength and Rigidity Reliability Requirements, Repeated Loads and Fatigue", MIL-A-8866 (USAF), August 1975.
5. Military Specification, "Airplane Strength and Rigidity Ground Tests", MIL-A-8867B (USAF), August 1975.
6. Pinckert, R. E., "Damage Tolerance Assessment of F-4 Aircraft", AIAA Paper 76-904, September 1976.
7. Stanford, R. D. and Morrow, J. W., "The F/FB-111 Durability and Damage Tolerance Assessment Program", Proceedings of Aircraft Structural Integrity Conference, Macon, Georgia, November 1984.
8. Tiffany, C. F., et al, "Analysis of USAF Aircraft Structural Durability and Damage Tolerance Notebook", Proceedings of Aircraft Structural Integrity Conference, Wright-Patterson AFB, Ohio, 1978.
9. Smith, S. H., et al, "Development of a Plan for Organic Damage Tolerance Analysis Capability", Battelle Columbus Laboratories, Columbus, Ohio, April 1981.
10. Coffin, M. D. and Tiffany, C. F., "New Air Force Requirements for Structural Safety, Durability, and Life Management", Journal of Aircraft, pp. 93-98, February 1976.
11. Christian, T. F., Jr., "Enhanced Aircraft Structural Maintenance Using Organic Depot Damage Tolerance Analysis", AIAA Paper 83-2450, October 1983.
12. Engle, R. M., Jr., "CRACKS, A Fortran IV Digital Computer Program for Crack Propagation Analysis", AFFDL-TR-70-107, Wright-Patterson AFB, Ohio, 1970.
13. Chang, J. R. and Engle, R. M., Jr., "Improved Damage Tolerance Analysis Methodology", AIAA Paper 83-0863, May 1983.
14. Gallagher, J. P., et al, "USAF Damage Tolerant Design Handbook: Guidelines for the Analysis and Design of Damage Tolerant Aircraft Structures", AFWAL-TR-82-3073, Wright-Patterson AFB, Ohio, May 1984.
15. Pao, Y. C. and Whaley, P. W., "Computer-Aided Review and Planning of Material Fatigue Life", AIAA Paper 84-0968, April 1984.
16. Rich, D. L. and Impellizzeri, L. F., "Fatigue Analysis of Cold Worked and Interference Fit Fastener Holes", Cyclic Stress-Strain and Plastic Deformation Aspects of Fatigue Crack Growth, ASTM STP 637, 1977.
17. Noor, A. K., Storaasli, O. O., and Fulton, R. E., "Impact of New Computing Systems on Computational Mechanics and Flight Vehicle Structures Technology" in The Influence of Large Scale Computing on Aircraft Structural Design, AGARD-R-706, Sienna, Italy, April 1984.
18. Berens, A. P., et al, "Handbook of Force Management Methods", AFWAL-TR-81-3079, Wright-Patterson AFB, Ohio, April 1981.
19. Gibson, M. B., "C-5 Aircraft Structural Integrity Program and Its Impact on Mission Readiness", Proceedings of Aircraft Structural Integrity Conference, Macon, Georgia, November 1984.
20. Engle, R. M., Jr. and Christian, T. F., Jr., "Maintenance Impact of Current Loads Recording Methodology on Crack Growth Based Individual Aircraft Tracking", AIAA Paper 84-2410, October 1984.
21. Wenneker, J. C. and McFarland, J. L., "F-15 Force Structural Maintenance Plan", Proceedings of Aircraft Structural Integrity Conference, Macon, Georgia, November 1984.
22. Veldman, R. J., "Airborne Data Acquisition Multifunction System (ADAMS)", Proceedings of Aircraft Structural Integrity Conference, Macon, Georgia, November 1984.
23. Archer, B. K., "Development and Implementation of a Standard Flight Recorder", Proceedings of Aircraft Structural Integrity Conference, Macon, Georgia, November 1984.
24. Elliott, W. R. and Christian, T. F., Jr., "Methodology to Better Predict Structural Maintenance Requirements for Individual Aircraft", AIAA Paper 84-2411, October 1984.

25. Lincoln, J. W., "Method for Computation of Structural Failure Probability for an Aircraft", ASD-TR-80-5035, Wright-Patterson AFB, Ohio, July 1980.
26. Saff, C. R. and Forness, S. D., "An Approach to Structural Risk Assessment Using Damage Tolerance Analysis and Flight Usage Data", Proceedings of Aircraft Structural Integrity Conference, Macon, Georgia, November 1984.
27. "Failure Analysis and Mechanisms of Failure of Fibrous Composite Structures", NASA-CP-2278, Langley Research Center, Virginia, March 1982.
28. "Characterization, Analysis and Significance of Defects in Composite Materials", AGARD-CP-355, London U.K., April 1983.
29. "Composite Structural Repairs", AGARD-R-716 and Addendum, Sienna, Italy, April 1984.
30. Schweinberg, W., et al, "Depot Level Repairability, Maintainability, and Supportability of Advanced Composites", AIAA Paper 83-2516, October 1983.
31. Pierce, J. D. and Okumura, W. E., "New Technologies - Their Impact on Air Force Depot Repair", Air Force Journal of Logistics, Vol IX, No. 1, pp 28-32, Winter 1985.
32. Whitehead, R. S., Ritchie, G. L., and Mullineaux, J. L., "Qualification of Primary Composite Aircraft Structures", Proceedings of Aircraft Structural Integrity Conference, Macon, Georgia, November 1984.
33. Demuts, E., Whitehead, R. S., and Deo, R. B., "Assessment of Damage Tolerance in Composites", Proceedings of Aircraft Structural Integrity Conference, Macon, Georgia, November 1984.

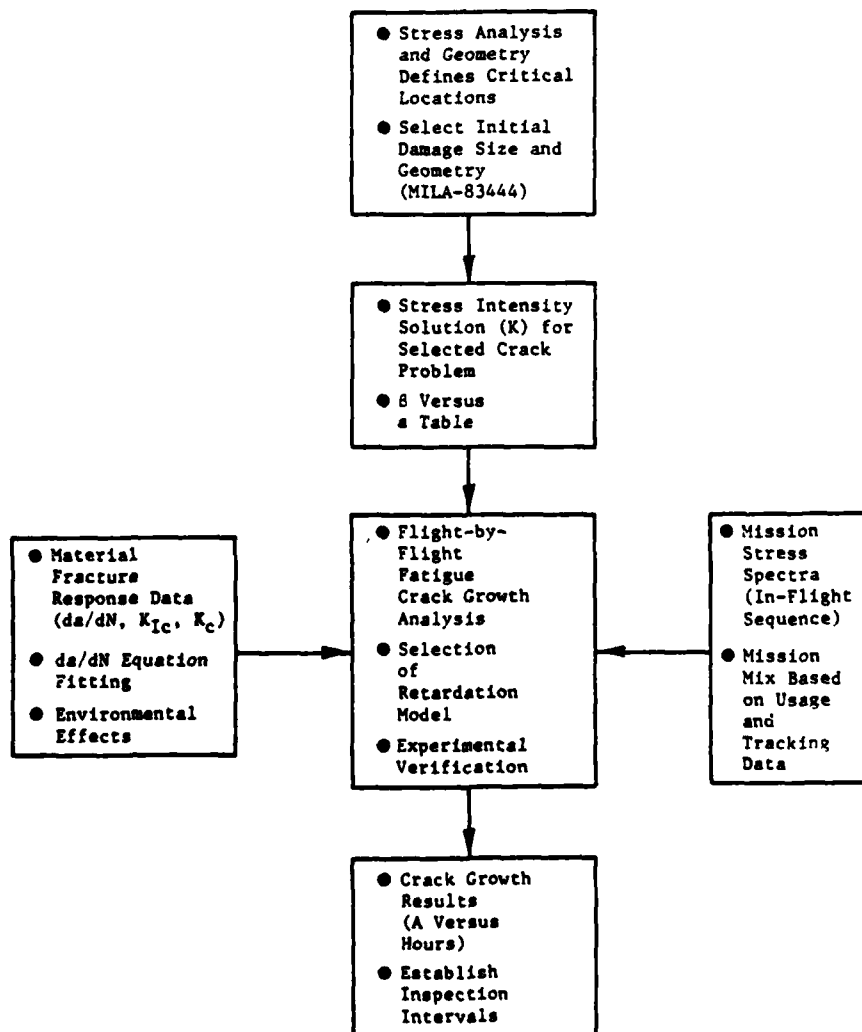


FIGURE 1. FLOW DIAGRAM OF SALIENT FEATURES OF FLIGHT-BY-FLIGHT FATIGUE CRACK GROWTH ANALYSIS

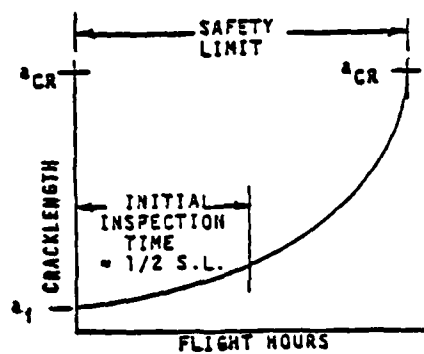


FIGURE 2 CRACK GROWTH CURVE

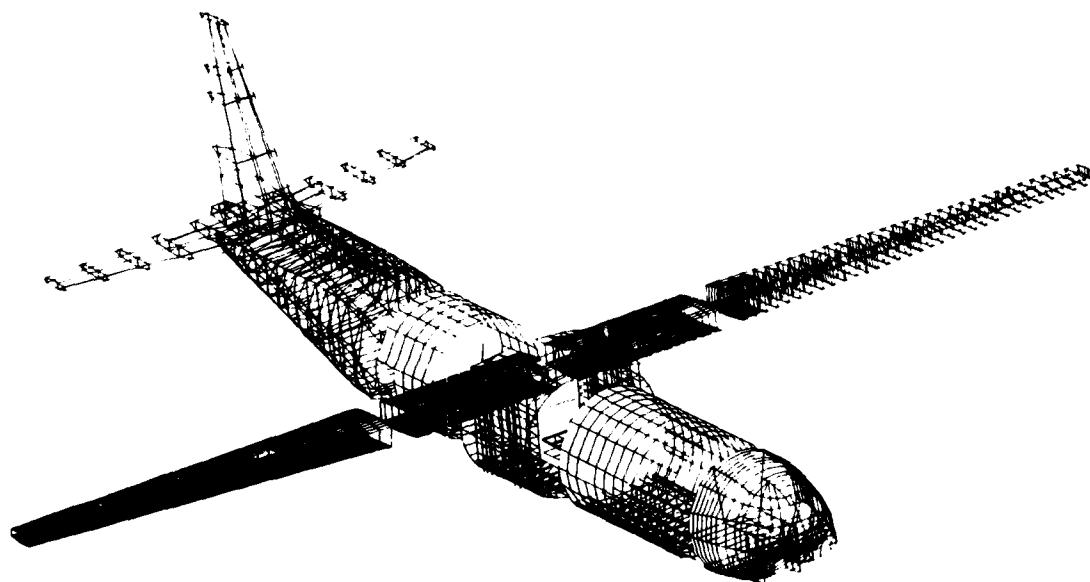


FIGURE 3 ENTIRE AIRFRAME FINITE ELEMENT MODEL

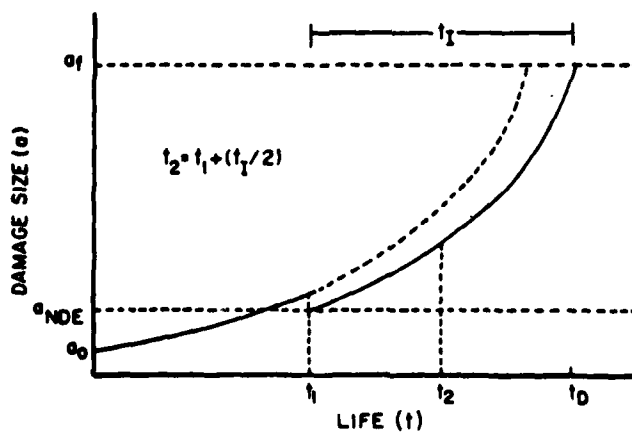


FIGURE 4 CRACK GROWTH RESET

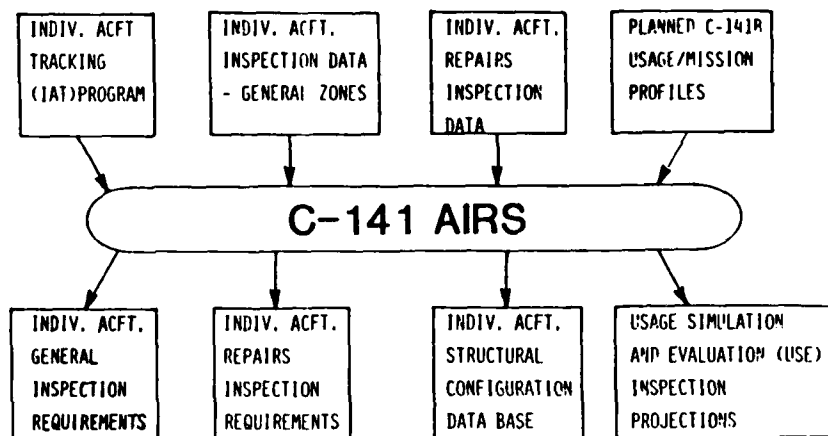
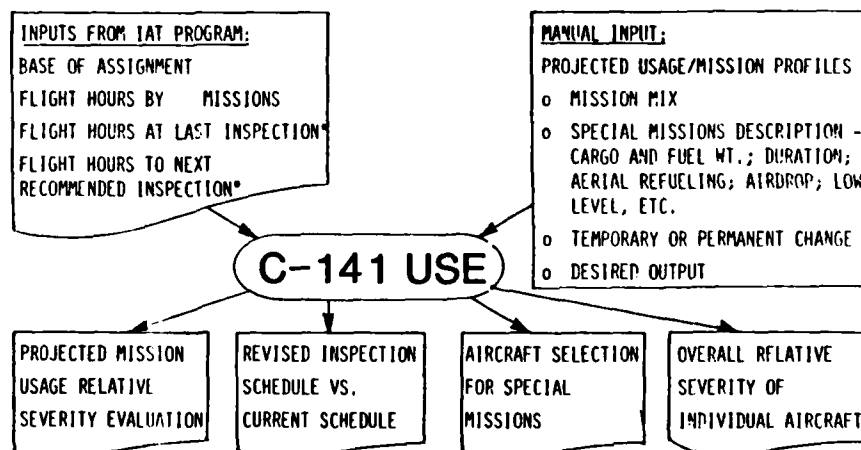


FIGURE 5 C-141 AIRS PROGRAM

## USAGE SIMULATION AND EVALUATION (USE) PROGRAM CONCEPT



\* FOR EACH SIGNIFICANT ZONE

FIGURE 6 C-141 USE PROGRAM



## IMPACT OF COMPUTER ADVANCES ON FUTURE FINITE ELEMENT COMPUTATIONS

Robert E. Fulton  
 George Washington University  
 NASA/Langley Research Center, Mail/Stop 269  
 Hampton, Virginia 23665

## SUMMARY

The design of complex aerospace vehicles such as advanced aircraft, spacecraft and space stations require continually increased levels of detail in supporting analyses. Such design activities require large order finite element models and excessive computation demands in both calculation speed and information management. Recent advances in data base management and supercomputer technology provide the opportunity to upgrade current structural analysis capabilities. Most existing major structural analysis software systems were designed ten to twenty years ago and have been optimized for current computers. Such systems often are not well structured to take maximum advantage of the recent and continuing revolution in computing capabilities such as distributed processing, information management, parallel processing and expert systems which are needed to support the design of the next generation of aerospace vehicles.

This paper discusses some research carried out over the past ten years in engineering data base management and parallel computing to suggest some opportunities for research toward the next generation of structural analysis capability. The paper is based on experiences in data base management associated with the IPAD project and in parallel processing associated with the FEM project, both sponsored by NASA. Using these results, several areas are identified as needed research to achieve a next generation structural analysis capability. Also included is a near term strategy for a distributed structural analysis capability based on relational data base management software and parallel computers which could serve as a baseline for evolving toward a future structural analysis system.

## INTRODUCTION

The design of complex aerospace vehicles such as advanced aircraft, spacecraft and space stations requires continually increased levels of detail in supporting analyses. For example, requirements make it necessary and technology advancements make it feasible to consider in dynamic analyses the combined effects of such things as composite materials, multidisciplinary interaction, three dimensional geometry and nonlinear behavior. Including such effects in complex structural problems may require large order finite element models and excessive computational demands in both calculating speed and information management. At the same time most structural analysis systems were designed ten to twenty years ago (Fig. 1) and have been optimized for current computers. Such systems often are not well structured to take maximum advantage of the recent and continuing revolution in computer capabilities in such areas as distributed processing, information management, parallel computation, and expert systems, which are needed to support the design of the next generation of aerospace vehicles (Fig. 2). The above vehicle requirements combined with the age of structural analysis systems suggest the need to advance the technology toward a new generation of structural analysis capability.

This paper discusses some research carried out over the past ten years in the two areas of engineering data base management and parallel computing to suggest some opportunities for research toward the next generation of structural analysis capability. The paper is based on experiences in data base management resulting from the IPAD (Integrated Programs for Aerospace-Vehicle Design) project (refs. 1-3) and in parallel processing resulting from the FEM (Finite Element Machine) project (ref. 39), both sponsored by NASA. Included is a brief summary of work from each project together with a proposed near term strategy for incorporating these capabilities into a software framework to serve as the baseline for a future structural analysis system.

## IPAD ENGINEERING DATA BASE MANAGEMENT APPROACH

A key to improved industry productivity is the effective management of engineering information. To stimulate advancement in this area a joint NASA/Navy/Industry project designated Integrated Programs for Aerospace-Vehicle Design (IPAD) was carried out from 1970-1984 with the goal of raising aerospace productivity through advancement of technology to integrate and manage information involved in the design and manufacturing process (Fig. 3). The IPAD research was guided by an Industry Technical Advisory Board (ITAB) composed of over 100 representatives from aerospace and computer companies (Fig. 4).

The IPAD project developed prototype computer software to meet many CAD/CAM information management requirements (refs. 1-4). Some of the basic requirements driving CAD/CAM systems development (refs. 5-21) include (Figure 5): (1) accommodate many different views of data from a variety of users and computing storage devices; (2) allow many levels of data descriptions to support a wide variety of engineering organizations and tasks; (3) permit easy changes in data definition as work progresses; (4) allow data to be distributed over networks of computers of various manufacture; (5) permit data definitions to be readily extended as needs arise; (6) store and manipulate geometry information; (7) embody adequate configuration management features; and (8) provide extensive capability to management information describing stored data. The IPAD approach taken is to conduct appropriate research and develop prototype software for a future network of computers (refs. 22-24). To provide the required CAD/CAM functionality, and yet meet software performance requirements, data base management is staged at two or more levels with different software capabilities needed for both the local (user) level and global (project) level (Fig. 6). With such a tiered data base management approach, today's inconvenient file-oriented procedures can be replaced by future procedures (Fig. 7) where convenient user languages efficiently create, store, manipulate, access, and control information in accordance with CAD/CAM requirements.

Prototype software was developed under the IPAD project at both the local and global levels. A system denoted Relational Information Management (RIM) was developed for local level data management. RIM is based on the highly flexible relational models which organize and manage engineering and scientific information according to tables and relationships among tables. Its features include interactive queries, report writer, and FORTRAN interface. RIM was first operational in 1979 and is now a mature system. In 1981, it served as a critical information management capability to support NASA investigations (ref. 25) of the integrity of 30,000 tiles on the space shuttle (Fig. 8). The success of RIM in such evaluations has led to its continued development and enhancement by government and industry (Fig. 9). A public Version 5.0 is available from COSMIC\* for CDC, IBM, DEC, UNIVAC, PRIME, and Harris computers. Commercial organizations have continued to enhance RIM and now provide compatible RIM derivative software (e.g., BCS/RIM and MicroRIM) and associated maintenance and support for such software operational on a wide range of computers (from personal computers to super computers). Commercial versions of RIM are being used extensively by industry (refs. 26-29), and attendant RIM User Groups have been established to coordinate mutual needs. NASA used RIM as the common data management system (Fig. 10) to integrate several engineering analysis programs (Fig. 11) through the development and application of a prototype system denoted Prototype Integrated Design System (PRIDE) (refs. 49, 50). Through use of the RIM data communication features the PRIDE system includes a distributed data base capability across a network of heterogeneous computers (Fig. 12).

IPAD research also led to development of a global data base management system denoted IPAD Information Processor (IPIP). The approach taken in IPIP was to provide the capability within one system to manage information composed of a wide variety of information structures including hierarchical, network, relational, and geometric. The IPIP approach uses multiple levels of information formats (schemata) to permit unlimited reorganization of information as work progresses (Fig. 13). Each schema is connected to other schemata via a general-purpose mapping capability (language). IPIP is a new concept, still in test and evaluation phases, and is currently operational on a CDC computer. Its approach to management of geometric data is a unique and important concept which could be very important to future integration of design and manufacturing. A critical technical challenge to IPIP development has been to provide the high degree of engineering user flexibility and yet achieve acceptable response times. In late 1983 a test system which has user responses for test problems of less than 0.5 seconds was provided to selected ITAB organizations to support its evaluation. IPIP has also been established by one computer vendor as an "as is" product and limited support is provided by the vendor for its installation and evaluation. IPAD results to date in defining CAD/CAM data management requirements and in developing prototype software have helped stimulate development of commercial CAD/CAM data management software (refs. 30-33), and several computer vendors plan release in 1984 of relational-type data management systems which address many of the CAD/CAM requirements identified in IPAD research. IPAD results have also helped stimulate infusion of data base management technology into university engineering research (refs. 34-36).

A critical CAD/CAM requirement not yet contained within any available or planned commercial data management system is the ability to efficiently manage geometry information in concert with other engineering data (Fig. 14). Through use of the multischema capability, IPIP provides the first approach to management of geometry information within a data management system (refs. 13, 14). The IPIP approach provides software capability to create on top of the basic geometric data an information structure having an unlimited number of geometric descriptions (schemata). One geometry schema includes the evolving geometry/graphics standard, Initial Graphics Exchange Specifications (IGES). This IPIP information structure concept opens the door for convenient integration of geometric information with other types of information associated with a CAD/CAM development process (Fig. 15). An evaluation of the IPIP geometry concept is now underway, and comparisons are being made with other approaches in which management of the geometric data takes place outside the basic data base manager.

#### CONCURRENT PROCESSING APPROACH FOR STRUCTURAL ANALYSIS

The complex structural analyses required for advanced aerospace vehicle design may require large order finite element models and very large scale computations. Design of such vehicles will require in some cases effective computer speeds greater than  $10^3$  MFLOPS (million floating point operations per second) for timely results. These speeds are well beyond the capabilities of current sequential computers.

Projected advances in computer technology indicate significant increases in effective calculation speed will be available in the 1990's through computer architectures consisting of arrays of processors operating concurrently on different tasks (refs. 37, 38). As indicated in Figure 16, such advanced computers denoted MIMD (multiple instruction, multiple data) computers have the potential for increasing effective calculation speeds by several orders of magnitude. The key to achieving this increased speed for structural analysis is the selection, development and implementation of appropriate numerical algorithms which take advantage of the concurrent computation features of this new generation of computers. Use of existing conventional algorithms and software will not realize the full potential of these new MIMD computers; new algorithms are needed which support structural calculations on concurrent computers.

Studies carried out under the FEM project have included investigations of several numerical analysis algorithms for MIMD computers and assessed the increase in computation speed relative to a sequential computer implementation. Two algorithms studied include eigenvalue determination and numerical integration methods which have application to free vibration and transient response analysis of complex finite element models. This section summarizes the methods, discusses concurrent implementation criteria, and provides results from application of the methods to test problems on an experimental MIMD computer.

#### Implementation on an Experimental MIMD Computer

The total hardware/software system must be taken into consideration when developing and implementing

\*Computer Software Management and Information Center, 112 Barrow Hall, University of Georgia, Athens, GA 30602.

a concurrent algorithm. The implementation criteria that influence the efficiency of an algorithm include the amount of computation versus the amount of communication in a given problem, the balance of workload among the processors, the communication paths and synchronization delays, and the size of a problem relative to the number of processors used. The flow of the algorithm must be analyzed to identify those calculations which must be sequential and those which can be done in parallel. Efficient algorithms typically maximize parallel calculations, minimize sequential calculations, minimize communication, and partition tasks onto a processor array so that the communication paths are most effective.

The computer hardware/software system used to support this study was the Finite Element Machine (FEM) (fig. 17), an experimental MIMD computer developed at the NASA Langley Research Center to support parallel processing research for structural applications (ref. 39). The FEM design is an array of microprocessors connected together by up to 12 local links plus a global bus (fig. 18). Each processor has its own memory and can be programmed to perform calculations independently. A small minicomputer acts as a front-end controller where programs are coded and compiled.

Through use of custom designed system software (refs. 40, 41), FEM programs are sent from the controller to the processor array. While the program is executing on the array, the software provides control and runtime monitoring. When execution is completed, output is sent back to the controller. The processors can run in two modes: synchronous or asynchronous. In the synchronous mode, input is queued in order in the receiving processor until it is used. Thus, if one processor needs information from another processor, execution is halted until such information is received. In an asynchronous mode, data sent to one processor can be written over by subsequent data. In some algorithms, this approach is desirable, as the new information is more meaningful. The FEM configuration used for this study was limited to eight asynchronous processors connected together by eight local links and by a global bus.

#### Inverse Power Method for Eigenvalue Analysis

The approach used for eigenvalue and eigenvector determination is the inverse power method tailored to take advantage of concurrent processing capabilities. The inverse power method is an iterative method often used for determining vibration frequencies and mode shapes for large order structural problems (ref. 42 and 43). It is particularly useful since it retains the original form of the stiffness matrix, its associated sparsity and its numerical accuracy, as well as any matrix decompositions previously obtained. For a brief summary of the method consider the following eigenvalue problem:

$$K X = \lambda M X \quad (1)$$

where  $K$  and  $M$  are the symmetric stiffness and mass matrices, respectively, and  $\lambda$  is the eigenvalue representing the square of the frequency. The inverse power method iterates through the sequence:

$$K V^{r+1} = \lambda M V^r \quad (2)$$

where  $r$  denotes the  $r$ th iteration and  $V^r$  the current estimate of the eigenvector. The method converges to the lowest  $\lambda$  and associated eigenvector. Larger eigenvalues and associated eigenvectors can be obtained through use of orthogonality to "sweep out" the effects of the lower eigenvectors from the  $V^r$ . For example, the method converges to the next lowest  $\lambda$  and associated eigenvector if the assumed vector  $V^r$  is taken as

$$V^r = V - c_1 X_1 \quad (3)$$

where  $V$  is an arbitrary vector and  $c_1$  is the participation of the first eigenvector  $X_1$  in  $V$  and is given by

$$c_1 = \frac{X_1^T M V}{X_1^T M X_1} \quad (4)$$

Convergence of the method can be assessed through comparison of all elements of the two vectors  $V^{r+1}$  to  $V^r$  or through comparison of their norms. The converged eigenvalue  $\lambda_1$  can be determined through use of the Rayleigh Quotient associated with the eigenvector  $X_1$ ; that is

$$\lambda_1 = \frac{X_1^T K X_1}{X_1^T M X_1} \quad (5)$$

The method can be used to obtain the  $\lambda$  closest to a reference value  $\beta_j$  (denoted a "shift parameter") by inserting a shift in the eigenvalue scale. In equation (2) let:

$$\lambda = \beta_j + \bar{\lambda} \quad (6)$$

which results in

$$(K - \beta_j M) V^{r+1} = \bar{\lambda} M V^r \quad (7)$$

In the concurrent implementation of the inverse power method, shifts are used and each processor calculates the eigenvalues in the neighborhood of a specified reference value  $\beta_j$ . Each processor then obtains a given number of eigenvalues in the range. An eigenvalue spectrum and the concurrent inverse power method mapped onto four processors are illustrated in Figure 19. The  $j$ th processor calculates the eigenvalues in the neighborhood of  $\beta_j$ . If the  $\beta_j$  are too close to each other, the results from the respective neighboring processors may overlap when several eigenvalues are obtained in each processor. There is a trade off between the number of eigenvalues to be calculated near a reference  $\beta_j$  value and the selection of a new  $\beta_j$ . The decision to select a new  $\beta_j$  or to calculate more eigenvalues for a given  $\beta_j$  can be controlled either by the user or by an appropriate executive algorithm. The user controlled

approach for decision making is summarized here and further details on an executive controlled approach are given in ref. 44. The major calculation steps are:

1. Assign to each processor the task of calculating eigenvalues near a specific reference eigenvalue.
2. Use the inverse power method with shifts to calculate a sequence of eigenvalues and associated eigenvectors closest to a reference value. Use the Rayleigh quotient to refine eigenvalue results.
3. Establish a control procedure for the concurrent calculations that is either: (1) user directed; or (2) directed through executive software which can be assigned to one of the processors as a part of its workload. Either approach should focus on tracking the eigenvalue results from the various processors, selecting parameter ranges and balancing processor workload.

For a user controlled approach, each processor executes the inverse power method independently and asynchronously. Each processor is given a different shift parameter  $\beta_j$  and assigned the task of solving for a given number of eigenvalues closest to  $\beta_j$  and associated eigenvectors. This approach is useful for the user who does not need all eigenvalues but wants eigenvalues in several specified ranges. Because there is no communication requirement among processors, a significant speedup is achieved compared to using the same algorithm sequentially. Each processor, however, takes a different computation time to solve for its assigned number of eigenvalues.

Total time to complete a solution phase is the time for the processor with the largest workload to complete its tasks. Since each processor works independently, there is no communication or synchronization overhead, but there is an imbalance in workload. Furthermore, there is no mechanism other than user insight to ensure against duplicate or redundant eigenvalue calculations. The approach could be useful in a multiprogramming environment where other user tasks could be performed on any idle processor. An executive controlled approach discussed in reference 8 provides a more convenient automated search strategy for balancing workload and determining eigenvalues within a range, but it also results in some increase in overhead.

#### Application of Concurrent Inverse Power Method

To evaluate the concurrent inverse power method, a test problem was solved on an array of FEM processors. The test problem was chosen to fit the memory constraints of the Finite Element Machine, and embody some of the aspects of large order structural vibration problems which impact concurrent implementation. For example, the stiffness matrix is sparse, there are groups of closely spaced frequencies, and the rate of convergence of the inverse power method varies over the frequency range.

The test problem (Fig. 20) deals with the flexural vibration of a long beam simply supported with 16 uniformly spaced supports. The beam mass is assumed to be lumped at the supports with a rotational inertia of  $m = 1.0$  at each support. The beam is modeled by 15 finite elements with one element between each support. Since the lateral displacements are constrained to be zero, the finite element degrees of freedom are the beam rotations at each of the 16 supports. The beam stiffness  $EI=1.0$  for all segments except the first where  $EI=2.0$ . The first three mode shapes and frequencies are shown in Fig. 21.

In the solution procedure, each processor was assigned the task of obtaining a specific number of eigenvalues in an eigenvalue range to attempt to balance the workload. Specifically the sixteen total eigenvalues sought were assigned equally among the working processors; e.g., two processors were assigned to find eight eigenvalues each, four to find four each, and eight to find two each. The solution times for each of the cases are given in Table 1. Since each processor works independently, there is no communication or synchronization overhead in the calculations. There is, however, an imbalance in the workload, as shown by the timing results in Table 1. For example, when only two processors are operating, each solving for eight eigenvalues, the first processor operates for 210 seconds while the second operates for 192 seconds. The sequential algorithm requires 402 seconds to solve the problem; thus there is a "speedup" in calculation time of 1.9., where speedup is the ratio of sequential calculation time (402 sec.) to concurrent computer calculation time (210 sec.). If processor workloads are balanced and all processors operate at maximum efficiency, the speedup is proportional to the number of processors. This result is denoted in the extreme right column of Table 1(a) as the theoretical limit for speedup. Measured computing "speedups" for the test problem are shown in Figure 22. The speedup results for the test problem appear representative of concurrent processing speedups achievable for large order eigenvalue problems through user-controlled task assignments for the inverse power methods. Other examples and improvements to the method are given in ref. 44.

#### Structural Dynamic Response Calculations

As a second example consider the dynamic response of a discrete structure subjected to transient forces. In general for a lumped mass formulation of a finite element or finite difference structural model, the acceleration for each degree of freedom for each structural node is a function of all displacements, velocities and time. In the concurrent computer implementation (ref. 45, 46), the  $m$  equations of motion can be distributed over  $n$  processors. A typical distribution of  $m$  equations of motion is

$$\begin{aligned} \ddot{w}_1 &= f_1(w_1, \dot{w}_1, \ddot{w}_1, \dots, w_m, \dot{w}_m, \ddot{w}_m, t) && \text{Processor 1} \\ \ddot{w}_2 &= f_2(w_1, \dot{w}_1, \ddot{w}_1, \dots, w_m, \dot{w}_m, \ddot{w}_m, t) \\ & \\ \ddot{w}_3 &= f_3(w_1, \dot{w}_1, \ddot{w}_1, \dots, w_m, \dot{w}_m, \ddot{w}_m, t) && \text{Processor 2} \\ \ddot{w}_4 &= f_4(w_1, \dot{w}_1, \ddot{w}_1, \dots, w_m, \dot{w}_m, \ddot{w}_m, t) \end{aligned}$$

$$\ddot{w}_i = f(w_1, w_2, \dots, w_i, \dot{w}_1, \dot{w}_2, \dots, \dot{w}_m, t) \quad \text{Processor } n$$

$$\ddot{w}_m = f_m(w_1, w_2, \dots, w_i, \dot{w}_1, \dot{w}_2, \dots, \dot{w}_m, t)$$

where  $w_i$  is the displacement of the  $i$ th degree of freedom and a dot denotes differentiation with respect to time. If  $m$  is a multiple of  $n$ , there is an equal distribution of the computational load. Otherwise, the work load should be judiciously balanced to minimize processor idle time and increased overhead.

A common procedure for integrating dynamic equations is the Newmark-Beta method (ref. 47). The original formulation does not use matrix inversion but calculates new accelerations iteratively. This formulation was chosen for this study to maximize parallelism of the algorithm. The  $m$  equations of motion are distributed and solved on  $n$  processors (i.e.,  $m=n$ ) as indicated in Figure 23. All processors perform identical functions (same software) and act as "integration engines." Moving from  $t$  to  $t+\Delta t$ , each processor begins with an assumed acceleration for each assigned degree of freedom. The corresponding first derivatives and displacements are calculated using the following equations

$$\dot{w}_i(t+\Delta t) = \dot{w}_i(t) + (\ddot{w}_i(t) + \ddot{w}_i(t+\Delta t)) t/2 \quad (1)$$

$$w_i(t+\Delta t) = w_i(t) + \Delta t \dot{w}_i(t) + \Delta t^2 (\ddot{w}_i(t) b + \ddot{w}_i(t+\Delta t) (0.5-b)) \quad (2)$$

where  $b=1/4$  (trapezoidal integration).

The computation is then interrupted so that each processor can communicate results for its assigned degrees of freedom to its neighboring processors. The accelerations are then computed and compared with the assumed accelerations. If a given convergence criterion for the assigned degrees of freedom is met, then "local convergence" is achieved. The computations are interrupted again to check for convergence of all processors (i.e., global convergence) by using a flag network that sets a flag on each converged processor and sends its status to other processors. If either the local or the global convergence test fails, each processor uses the current calculated value as the assumed acceleration and repeats the computations. When global convergence is met, all processors simultaneously proceed to the next time step.

The concurrent solution algorithm was applied to sample dynamic response problems, the finite element analysis of a two-dimensional beam grillage under a uniform step load at its nodal points and simply supported at the boundaries (Fig. 24). Typical displacement responses are shown on the lower right of Figure 24. The primary computational results of interest are the times to calculate the structural response on a varied number of processors. The computational speedup derived from the concurrent processing approach compared to the sequential approach can then be computed. The computational speedup is defined as the computation time to calculate results on one processor divided by the computation time to calculate the same results on  $n$  processors.

The computational speedups versus the number of processors for the beam grillage problem is shown in Figure 25. The theoretical maximum speedups would be the speedups if there were no overhead for concurrent processing. Thus, the theoretical computational speedups are equal to the number of processors used. The computational speedups for solving the beam grillage problem using 24 and 48 equations are shown in Figure 9. For this example, there is a speedup of up to 7.36 using eight processors. The lack of maximum efficiency in the speedup is due to several factors. For example, since the transferral and receipt of data are not instantaneous, communication is one of the factors that contributes to the overhead. For iterative methods, it is necessary to synchronize the processors before making convergence checks, which creates another source of overhead. In addition, there is overhead from establishing looping parameters and indices identifying processors uniquely and from nonparallel coding. The results suggest that future concurrent processing computers hold the promise for significant speedups in computing capabilities to support large order nonlinear dynamic analyses (Fig. 26).

#### PROPOSED STRATEGY FOR FUTURE STRUCTURAL ANALYSIS SYSTEMS

The results of refs. 37 and 48 indicate that future structural analysis software systems should be restructured to be modular in algorithms, software structure and hardware implementation (Fig. 27). Unfortunately, such an approach is a long-term solution which requires redesign and coding of a software base at the same time the hardware base is changing rapidly and alternatives are growing. Research is needed in many areas to facilitate the development (Fig. 28).

A strategy which is based on an extension of the PRIDE distributed data base approach may be a useful approach to facilitate evolution to the future. The key is the data base management approach. Figure 29 illustrates an approach where the key integrator is a multimachine flexible data base management capability such as RIM. This concept recognizes the need for a wide range of computing processors to support engineering calculations. The processors may range from workstations to mainframes and include widely different sequential, concurrent, and/or symbolic processors. On each processor is installed a common relational type data base manager and executive which have the same user functions and information format. The user controls transfer of information among processors as well as the sequence of major calculation steps. Each processor then carries out its functions and/or applications as appropriate. Data is moved among processors conveniently in a standard user-oriented format and transformed to other required formats as required. The relational data base management features facilitate the data transformation and existing application programs are retained on the processor best suited for their use. New applications which may be symbolic, sequential, and/or concurrent processor based can be developed and integrated into the system as appropriate and modeling and/or graphics capabilities generated on other processors can be easily used.

This approach is easily implementable and provides a good strategy for using existing application programs in both structural analysis and other disciplines. As upgraded, more modular applications software systems are developed they can be incrementally added to the system in an evolutionary fashion. As relational type data base management systems evolve with improved performance, these can be readily integrated into the total system. As new hardware evolves, the key new step is the installation of the appropriate executive and data base manager on the new hardware.

## CONCLUDING REMARKS

The design of complex aerospace vehicles such as advanced aircraft, spacecraft and space stations requires continually increased levels of detail in supporting analyses. Such design activities require large order finite element and/or finite difference models and excessive computation demands in both calculation speed and information management. Recent advances in data base management and supercomputer technology provide the opportunity to upgrade current structural analysis capabilities. Most existing structural analysis software systems were designed ten to twenty years ago and have been optimized for current computers. Such systems often are not well structured to take maximum advantage of the recent and continuing revolution in computing capabilities such as distributed processing, information management, parallel processing and expert systems which are needed to support the design of the next generation of aerospace vehicles.

This paper discusses some research carried out over the past ten years in engineering data base management and parallel computing to suggest some opportunities for research toward the next generation of structural analysis capability. The paper is based on experiences in data base management resulting from the IPAD project and in parallel processing resulting from the FEM project, both sponsored by NASA. Using the results of these projects, several areas are identified as needed research to achieve a next generation structural analysis capability. Also included is a near term strategy for a distributed structural analysis capability based on relational data base management software and parallel computers which could serve as a baseline for a future structural analysis system.

## REFERENCES

1. IPAD: Integrated Programs for Aerospace-Vehicle Design. 1980, NASA CP-2143.
2. Fulton, Robert E., "Overview of Integrated Programs for Aerospace-Vehicle Design (IPAD)." 1980, NASA TM 81874.
3. Fulton, Robert E. and Brainin, Jack, "Management of CAD/CAM Information--Key to Improved Manufacturing Productivity." July 1984, NASA TM 86255.
4. Fulton, Robert E., "Critical Technology Issues for Integrated CAD/CAM." Proceedings of the Third Annual Conference and Exposition of the National Computer Graphics Association, Inc., Vol. 1. Washington, D.C.: Nat. Computer Graphics Assoc., Inc., 1982, pp. 326-335.
5. Birchfield, Burt, and Downey, Peter, "Product Definition Data Interface." IPAD II - Advances in Distributed Data Base Management for CAD/CAM, 1984, NASA CP-2301.
6. DeJean, Jean Pierre, "Integrated Information Support System (System Overview)." General Electric Paper presented at IPAD II National Symposium, Denver, CO, April 17-19, 1984.
7. "Manufacturing-Data Usage at Grumman - A Study of the Fabrication Phase," Grumman Aerospace Report, June 1983, prepared under Contract Y4D5803-PB3JN.
8. "IPAD II - Advances in Distributed Data Base Management for CAD/CAM." 1984, NASA CP-2301.
9. Comm. Navy Shipbuilding Technol., Marine Board, "Productivity Improvements in U.S. Naval Shipbuilding." 1982, Contract N00024-82-C-5349, Comm. Eng. & Tech. Sys., Natl. Res. Council., p. 32.
10. Colladay, R. S., "NASA Needs for Engineering/Scientific Data Management Technology. NASA paper presented at IPAD II National Symposium, Denver, CO, April 17-19, 1984.
11. Dube, R. Peter, and Johnson, H. Randall, "Computer-Assisted Engineering Data Base." Presented at ASME Winter Annual Meeting, Boston, MA, November 13-18, 1983. ASME Paper 83-WA/Aero-11.
12. McInnis, John W., "The Role of CAD/CAM Data Management in Navy Manufacturing Technology." U.S. Navy paper presented at IPAD II National Symposium, Denver, CO, April 17-19, 1984.
13. Shoosmith, J. M., and Fulton, R. E. (Guest Editors), "Geometric Modeling." Special Editions of Computer Graphics and Applications, Oct./Nov. 1983.
14. Dube, R. Peter, and Smith, Marcy Rivers, "Managing Geometric Information with a Data Base Management System." Special Edition of Computer Graphics and Applications on Geometric Modeling, Oct. 1983, pp. 57-62.
15. Ford, S. Jeane, "CIM's Bridge From CADD to CAM - Data Management Requirements for Manufacturing Engineering." IPAD II - Advances in Distributed Data Base Management for CAD/CAM, 1984, NASA CP-2301.
16. Barnes, Robert D., "Technology for the Product & Process Data Base." IPAD II - Advances in Distributed Data Base Management for CAD/CAM, 1984, NASA CP-2301.
17. Baur, R. G., Donthnier, M. L., Moran, M. C., Mortman, I., and Pinter, R. S., "Materials Properties Data Base Computerization." IPAD II - Advances in Distributed Data Base Management for CAD/CAM, 1984, NASA CP-2301.
18. Bohn, Pierre, "CAD/CAM and Scientific Data Management at Dassault." IPAD II - Advances in Distributed Data Base Management for CAD/CAM, 1984, NASA CP-2301.
19. Neuhold, Erich J., "Distributed Data Base Systems with Special Emphasis Toward FOREL." IPAD II - Advances in Distributed Data Base Management for CAD/CAM, 1984, NASA CP-2301.

20. Piscitelli, H., "Data Base Design and Implementation in Multiplant and Multi-Company Corporations." Goodyear paper presented at IPAD II National Symposium, Denver, CO, April 17-19, 1984.
21. Mitchell, Mary J. and Barkmeyer, Edward J., "Data Distribution in the NBS Automated Manufacturing Research Facility." IPAD II - Advances in Distributed Data Base Management for CAD/CAM, 1984, NASA CP-2301.
22. McKenna, E. G., "Design versus Manufacturing Data Base Management Requirements." IPAD II - Advances in Distributed Data Base Management for CAD/CAM, 1984, NASA CP-2301.
23. Balza, R. M., Beaudet, R. W., and Johnson, H. R., "A Distributed Data Base Management Facility for the CAD/CAM Environment." IPAD II - Advances in Distributed Data Base Management for CAD/CAM, 1984, NASA CP-2301.
24. Bryant, W. A., and Smith, M. R., "Data Management for Computer-Aided Engineering (CAE)." IPAD II - Advances in Distributed Data Base Management for CAD/CAM, 1984, NASA CP-2301.
25. Giles, Gary L., and Vallas, Maria, "Use of an Engineering Data Management System in the Analysis of Space Shuttle Orbiter Tiles." 1981, NASA TM 83215.
26. Breitbart, Yuri J., and Hartweg, Larry R., "RIM as an Implementation Tool for a Distributed Heterogeneous Database." IPAD II - Advances in Distributed Data Base Management for CAD/CAM, 1984, NASA CP-2301.
27. Karr, Patricia H., and Wilson, David J., "RIM as the Data Base Management System for a Material Properties Data Base." IPAD II - Advances in Distributed Data Base Management for CAD/CAM, 1984, NASA CP-2301.
28. Smith, Maurice, "Integrating Heterogeneous Nongeometric Information Systems Using RIM." Bendix paper presented at IPAD II National Symposium, Denver, CO, April 17-19, 1984.
29. Wray, William O., Jr., "A Wind Tunnel Database Using RIM." IPAD II - Advances in Distributed Data Base Management for CAD/CAM, 1984, NASA CP-2301.
30. Feldman, Harley D., "Control Data ICEM - A Vendor's IPAD-Like System." IPAD II - Advances in Distributed Data Base Management for CAD/CAM, 1984, NASA CP-2301.
31. Hartzband, D., "IPAD Influence on Digital Equipment Corporation." Digital Equipment paper presented at IPAD II National Symposium, Denver, CO, April 17-19, 1984.
32. Gingerich, J. Z., "Future Developments of a Hybrid CAD/CAM System." IPAD II - Advances in Distributed Data Base Management for CAD/CAM, 1984, NASA CP-2301.
33. Westervelt, Frank, "High Performance Data Base Systems for CAD/CAM CIM Applications." SMC Technology paper presented at IPAD II National Symposium, Denver, CO, April 17-19, 1984.
34. Plummer, Otho R., "The Database Management System: A Topic and a Tool." IPAD II - Advances in Distributed Data Base Management for CAD/CAM, 1984, NASA CP-2301.
35. Fenves, Steven J., "The Role of DBMS in Design Research." IPAD II - Advances in Distributed Data Base Management for CAD/CAM, 1984, NASA CP-2301.
36. Leach, Lanse M., and Wozny, Michael J., "The Impact of IPAD on CAD/CAM Database University Research." IPAD II - Advances in Distributed Data Base Management for CAD/CAM, 1984, NASA CP-2301.
37. Noor, A. K., Storaasli, O. O., and Fulton, R. E., "Finite Element Technology in the Future." Impact of New Computations on Computational Mechanics, (Noor and Pilkey, Editors), ASME Special Publication H00275, November 1983, pp. 1-32.
38. Hockney, R. W. and Jesshope, C. R., Parallel Computers, 1981, Bristol, Great Britain: Adam Hilger Ltd.
39. Storaasli, O. O., Peebles, S. W., Crockett, T. W., Knott, J. D. and Adams, L., "The Finite Element Machine: An Experiment in Parallel Processing." 1982, NASA TM 84514.
40. Crockett, T. W., PASLIB Programmer's Guide for the Finite Element Machine. December 1983, NASA CR-172281.
41. Knott, J. D., "FEM Array Control Software Users Guide." August 1983, NASA CR-172189.
42. The NASTRAN Theoretical Manual. NASA SP-221(06). January 1981, pg. 10.4-1.
43. Wilkinson, J. H., The Algebraic Eigenvalue Problem, 1965, Oxford, Great Britain: Clarendon Press.
44. Bostic, Susan W., and Fulton, Robert E., "A Concurrent Processing Implementation for Structural Vibration Analysis," presented at AIAA 26th SDM Conference (AIAA Paper 85-0783-CP), Orlando, Fla., April 15-17, 1985.
45. Storaasli, O., Ransom, J., and Fulton, R., Structural Dynamic Analysis on a Parallel Computer: the Finite Element Machine, May 1984, AIAA Publication 84-0966.

46. Ransom, J., Storaasli, O., and Fulton, R., "Application of Concurrent Processing to Structural Dynamic Response Computations." October 1984, NASA CP 2335.
47. Newmark, N., A Method of Computation for Structural Dynamics, ASLE Proceedings, Vol. 85, No. EM-3, July 1959, pp. 67-94.
48. Noor, Ahmed K., Storaasli, O. O., and Fulton, R. E., "Impact of New Computing Systems on Computational Mechanics and Flight-Vehicle Structures Technology," in AGARD Report 706, The Influence of Large Scale Computing on Aircraft Structural Design, April 2-6, 1984.
49. Fishwick, P. A., and Blackburn, C. L., "Managing Engineering Data Bases: The Relational Approach." Computers in Mechanical Engineering, Vol. 1, No. 3, January, 1983, pp. 8-16.
50. Blackburn, C. L., Storaasli, O. O., and Fulton, R. E., "The Role and Application of Data Base Management in Integrated Computer Aided Design." J. Aircraft, Vol. 20, No. 8, August 1983, pp. 717-725.

### TIMING RESULTS FOR FLEXURAL VIBRATION TEST PROBLEM:

User controlled calculations

(16 Eigenvalues)

Number of eigenvalues per processor	Calculations times (sec) for processor								Sequential time (sec)	Speedup*	Theoretical speedup limit
	1	2	3	4	5	6	7	8			
8	210	192							402	1.9	2.0
4	57	59	41	58					215	3.65	4.0
2	25	14	13	13	14	16	16	25	136	5.4	8.0

$$*\text{Speedup} = \frac{\text{Sequential time}}{\text{Maximum time in parallel}}$$

Table 1. Timing Results for Flexural Vibration Test Problem (16 Eigenvalues)

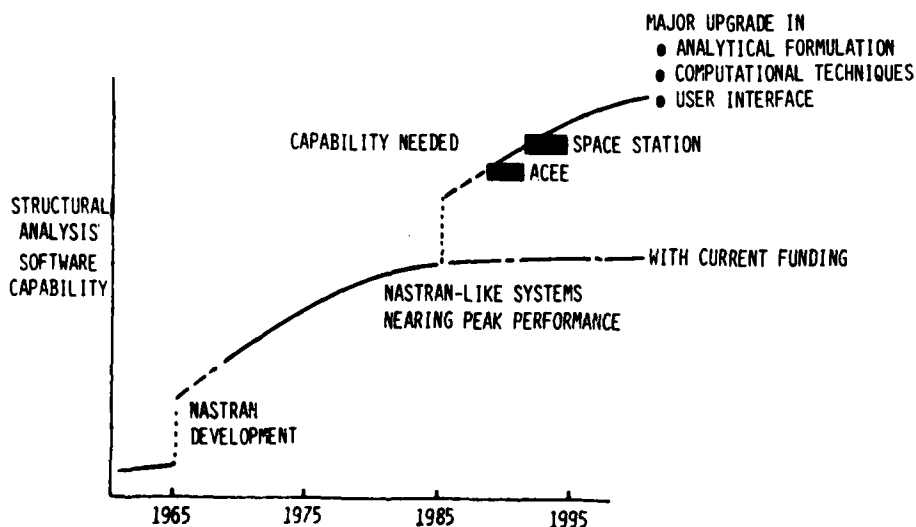


Figure 1. Historical Growth of Structural Analysis Systems



## 'New Developments' - Do We Need Them?

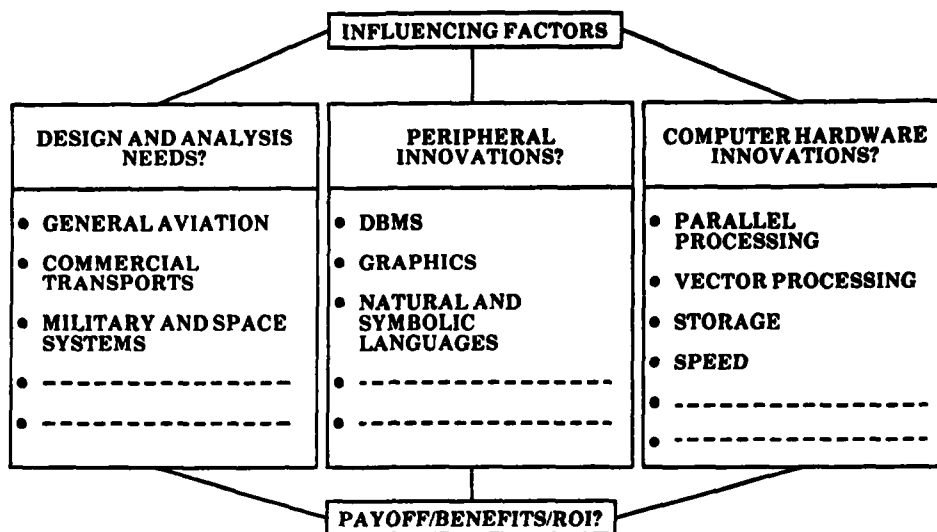


Figure 2. Factors Influencing Development of New Structural Analysis Systems

## JOINT NASA/NAVY RESEARCH PROGRAM TO DEVELOP TECHNOLOGY TO MANAGE CAD/CAM INFORMATION

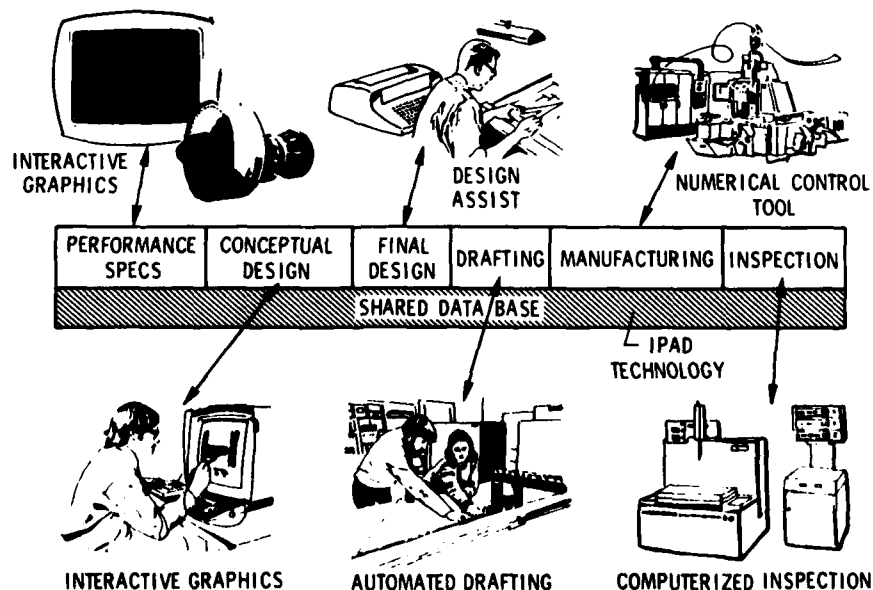
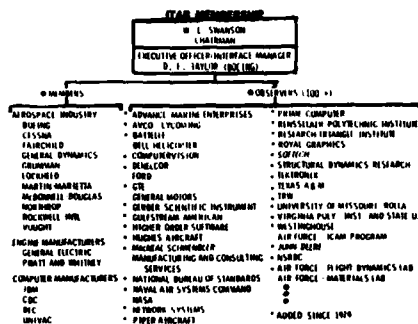


Figure 3. Joint NASA/NAVY Research Program to Develop Technology to Manage CAD/CAM Information

## IPAD INDUSTRY TECHNICAL ADVISORY BOARD (ITAB)



### ITAB ACTIVITIES

GUIDE DEVELOPMENT TO MEET INDUSTRY NEEDS

REVIEW/CRITIQUE ONGOING WORK

EVALUATE PROTOTYPE SOFTWARE

USE IPAD TECHNOLOGY AND PRODUCTS TO SPUR IN-HOUSE PLANNING AND DEVELOPMENT

Figure 4. IPAD Industry Technical Advisory Board (ITAB)

## CAD/CAM DATA MANAGEMENT SYSTEM REQUIREMENTS

- ACCOMMODATE MULTIPLE VIEWS OF DATA
- ALLOW MULTIPLE LEVELS OF DATA DESCRIPTION
- PERMIT CHANGES AND EXTENSIONS IN DATA DEFINITION
- DISTRIBUTE DATA OVER COMPUTER NETWORKS
- MANAGE GEOMETRY DATA
- PROVIDE CONFIGURATION MANAGEMENT CAPABILITIES
- MANAGE INFORMATION ABOUT DATA

Figure 5. CAD/CAM Data Management Systems Requirements

## IPAD APPROACH TO ENGINEERING DATA MANAGEMENT

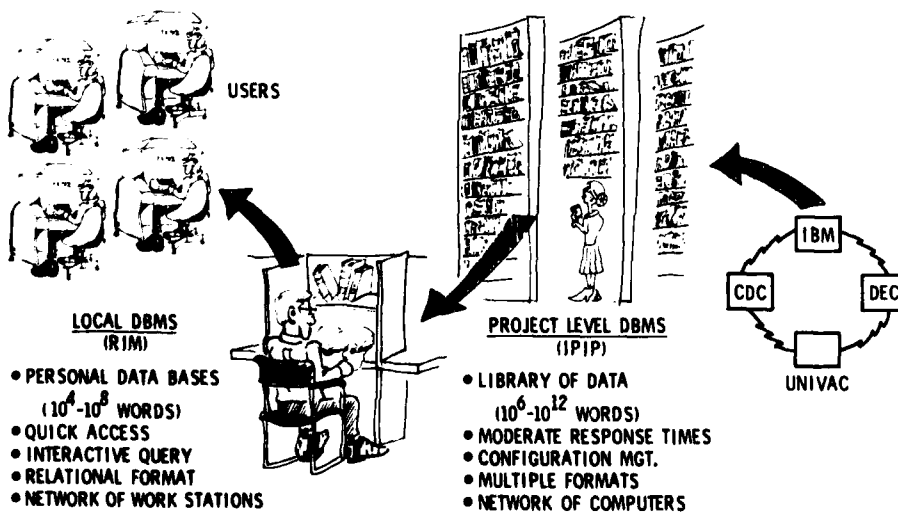


Figure 6. IPAD Multilevel Approach to Engineering Data Base Management

## HOW ENGINEERING GROUPS WILL USE IPAD INFORMATION INTEGRATION AND CONTROL

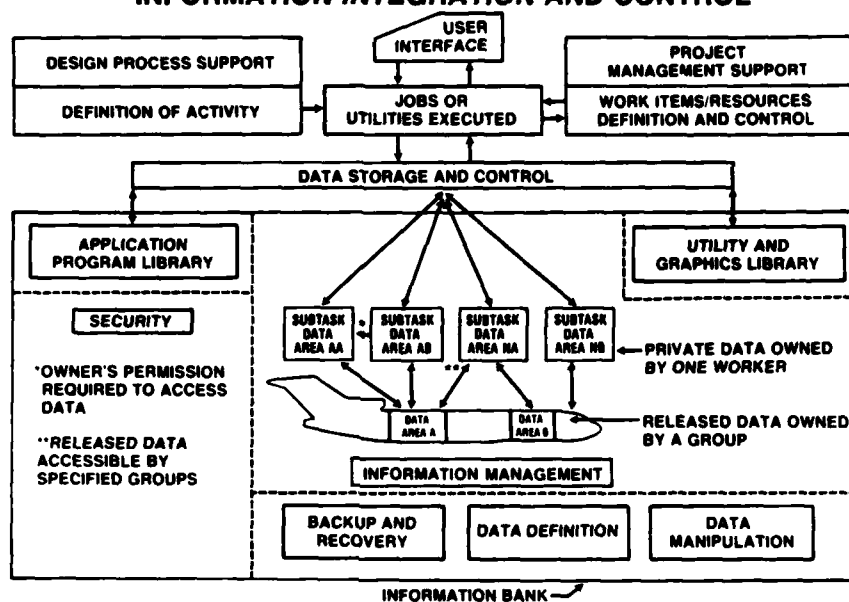


Figure 7. Engineering Use of Integrated CAD/CAM Data Management System

## USE OF IPAD/RIM DATA MANAGER TO SUPPORT INVESTIGATION OF SPACE SHUTTLE TILE ANALYSES

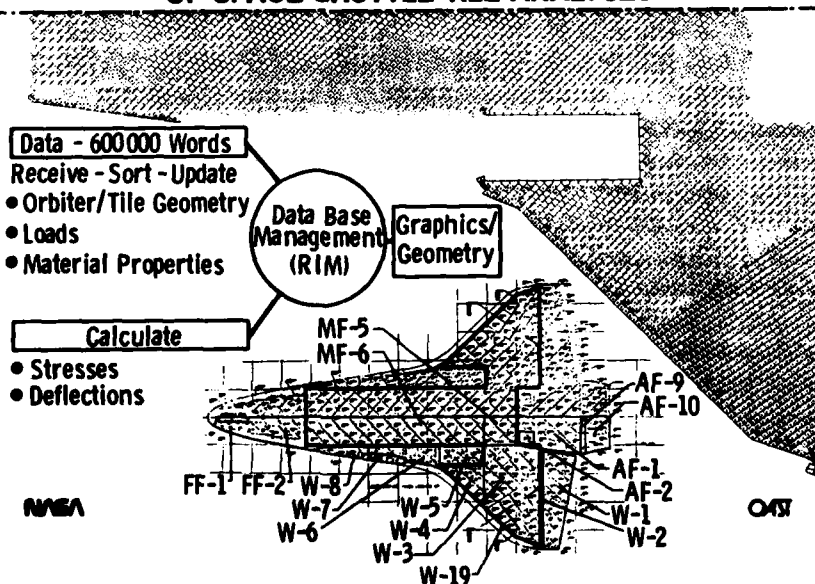


Figure 8. Use of IPAD/RIM Data Manager to Support Investigation of Space Shuttle Tile Analyses

## RIM STATUS

- WRITTEN IN FORTRAN
- CODE HIGHLY PORTABLE (CDC, VAX, PRIME, IBM, UNIVAC, CRAY, MICROS, ETC.)
- BASED ON RELATIONAL ALGEBRA
- FLEXIBLE QUERY LANGUAGE
- APPLICATION PROGRAM INTERFACE CAPABILITIES
- RIM TO RIM COMMUNICATIONS FILE
- SCHEMA CAN BE MODIFIED AS NEEDED
- VARIABLE LENGTH ATTRIBUTES
- COMMERCIAL VENDORS SUPPORTING/EXPANDING RIM

Figure 9. Features of RIM Relational Information Management System

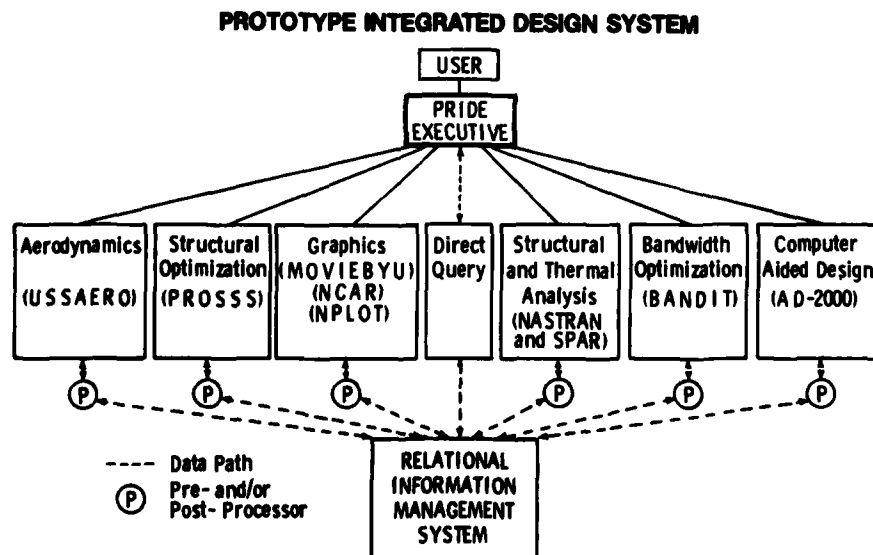


Figure 10. Organization of PRIDE Prototype Integrated Design System

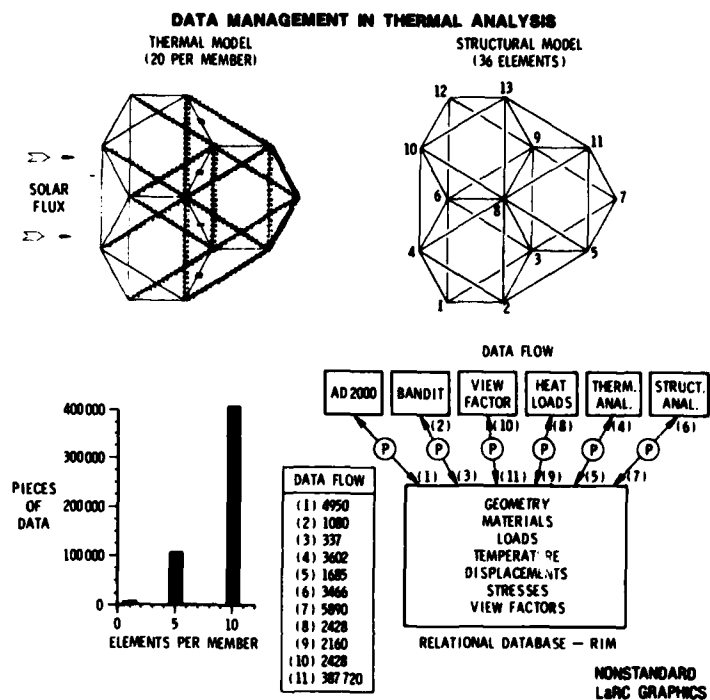


Figure 11. Application of PRIDE System to Thermal/Structural Analysis

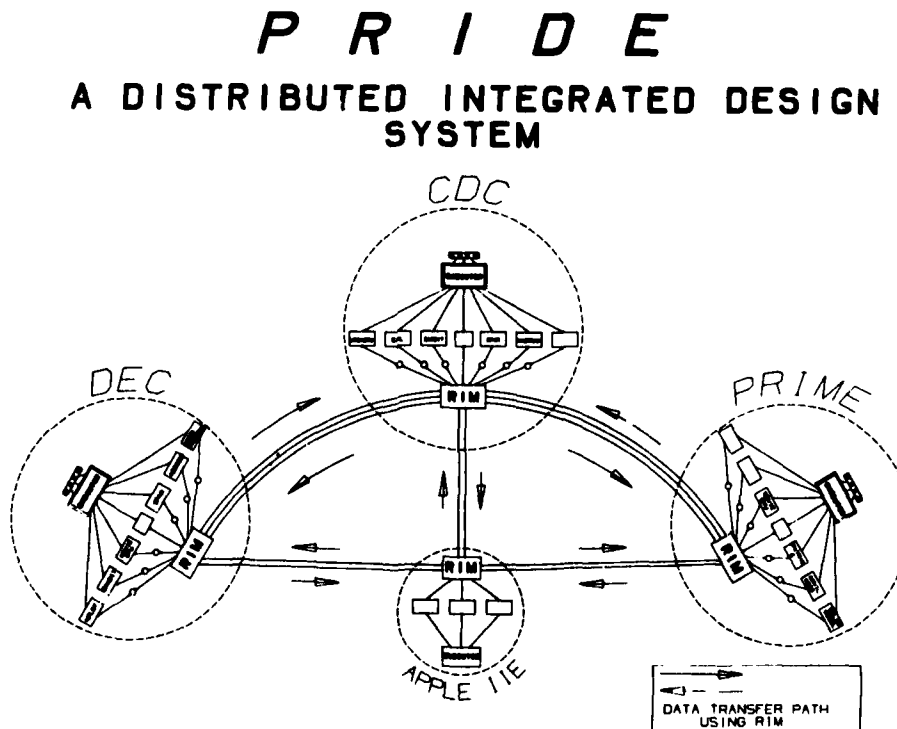


Figure 12. Application of PRIDE as a Distributed Integrated Design System

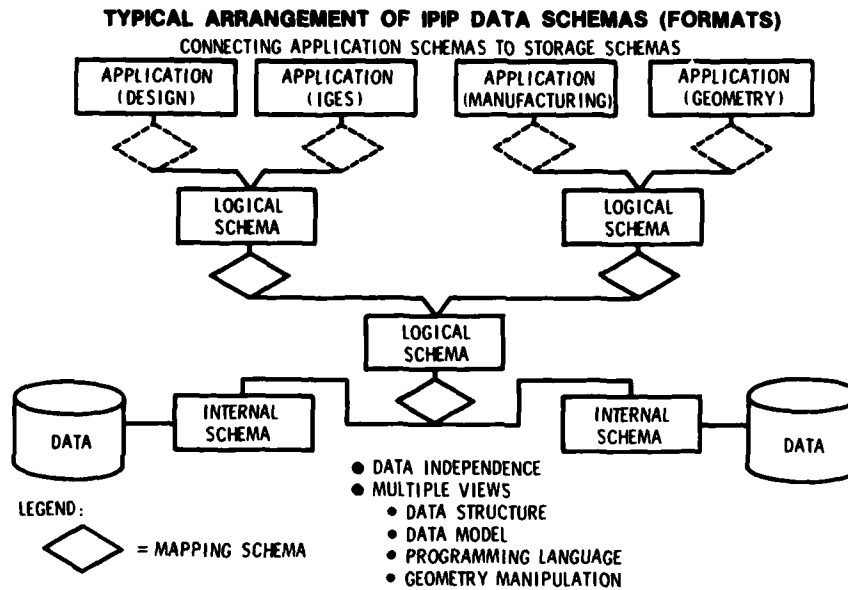


Figure 13. Typical Arrangement of IPIP Data Schemata (Formats) to Connect Application Schemata to Storage Schemata

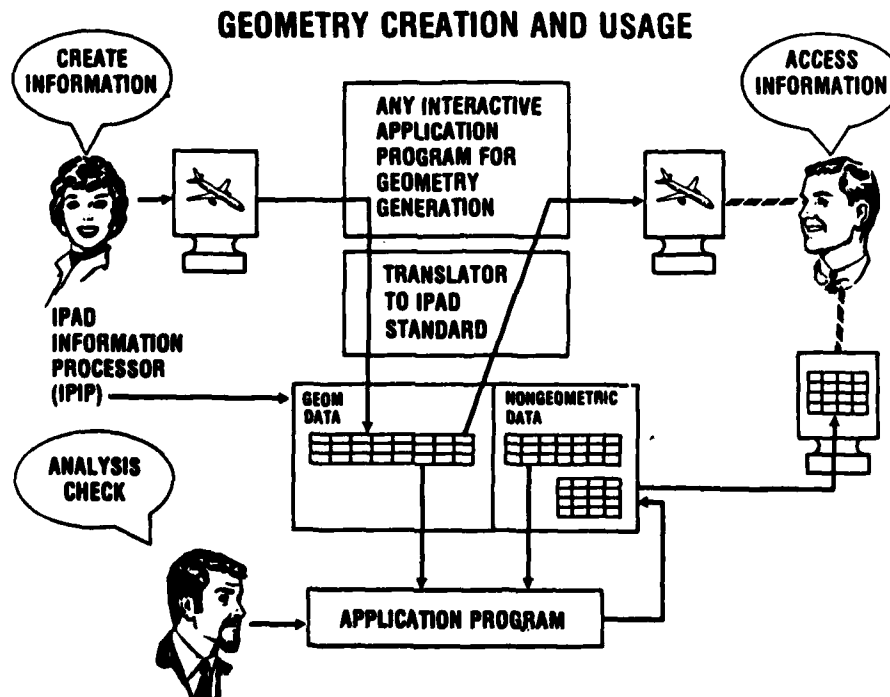
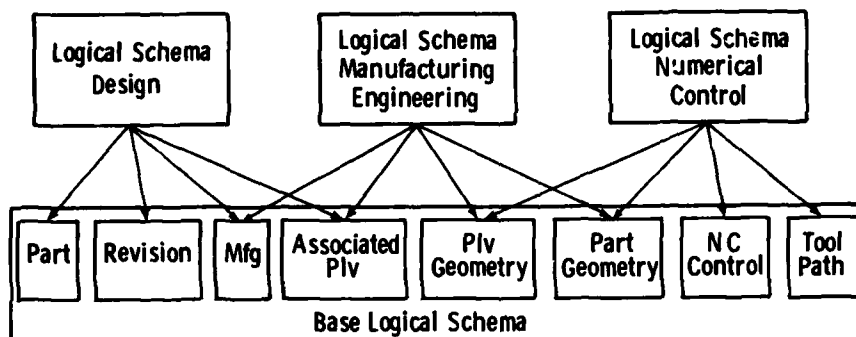


Figure 14. IPAD Approach to Unified Management of Combined Geometry and Non-Geometric CAD/CAM Data

# TYPICAL APPLICATIONS OF IPIP MULTISCHEMA CAPABILITY



## Integration of Design/Manufacturing Information

Figure 15. Typical Applications of IPIP Multischema Capability

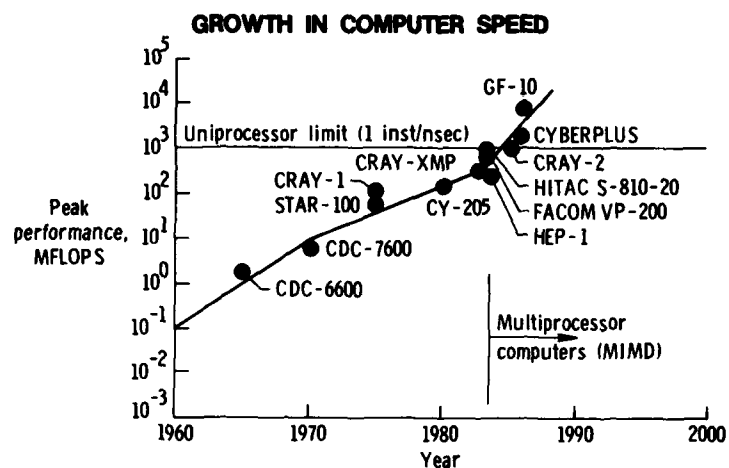


Figure 16. Growth in Computer Speed

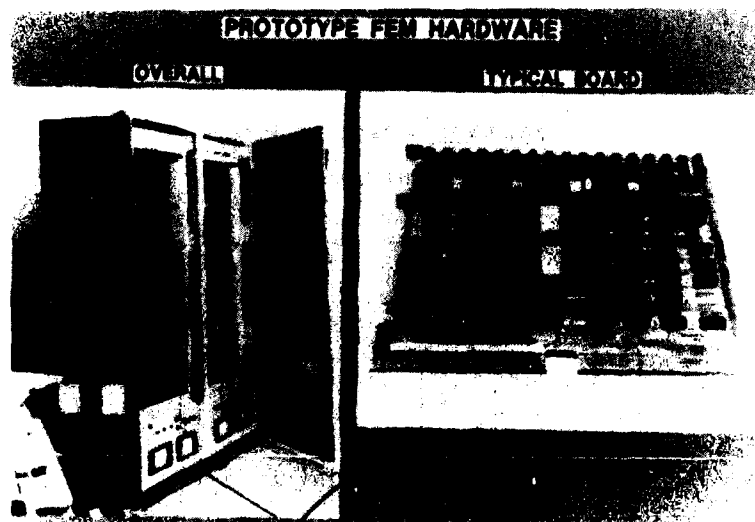


Figure 17. Finite Element Machine and Typical Board

## FEM HARDWARE AND SOFTWARE

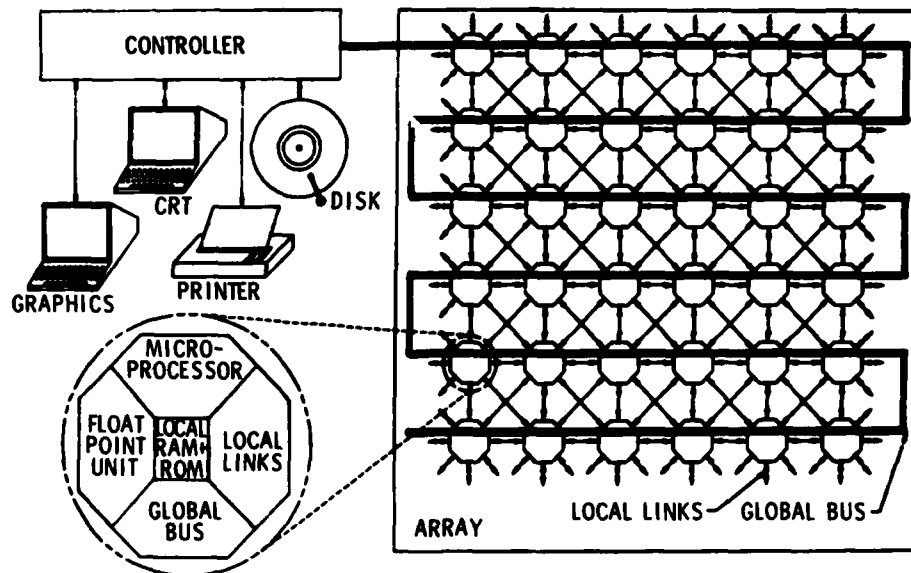


Figure 18. FEM Hardware and Software Organization

## CASE 1: PROCESSORS WORKING INDEPENDENTLY

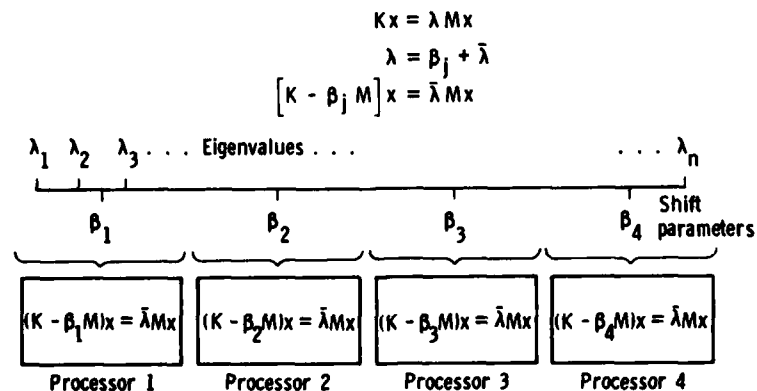
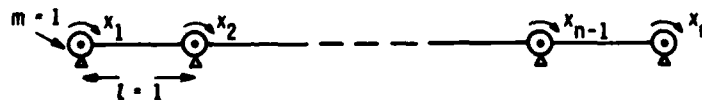


Figure 19. Four Processors Working Independently

## EIGENVALUE TEST PROBLEM

Vibration Multi-segment Beam



Eigenvalue Equations

$$\begin{bmatrix} 8 & 4 & & & \\ 4 & 12 & 2 & & \\ 0 & 2 & 8 & 2 & \\ & & \ddots & \ddots & \\ \circ & & 2 & 8 & 2 \\ & & & 2 & 4 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix} = \lambda \begin{bmatrix} 1 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ \circ & & & & 1 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix}$$

$$Kx = \lambda Mx$$

Figure 20. Test Problem



# REPRESENTATIVE FLEXURAL VIBRATION MODES

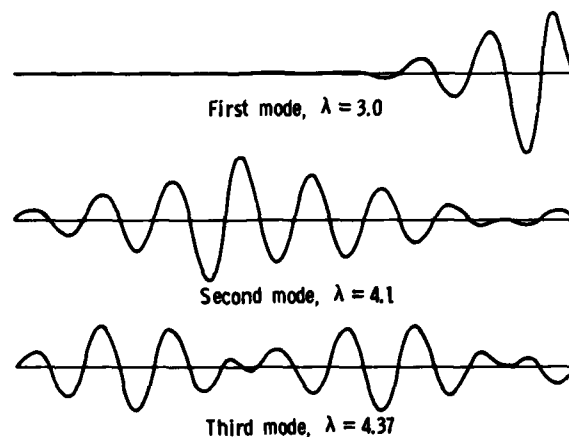


Figure 21. First Three Mode Shapes and Frequencies

## CASE 1: PROCESSORS WORKING INDEPENDENTLY

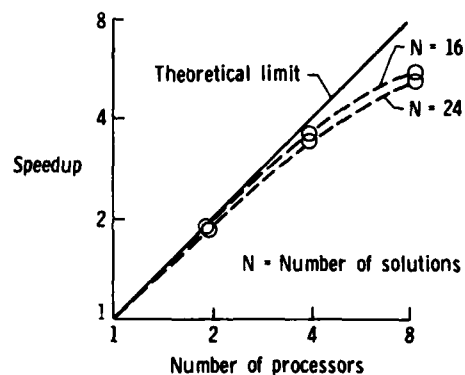


Figure 22. Parallel Processing Speedups

## PARALLEL INTEGRATION METHOD

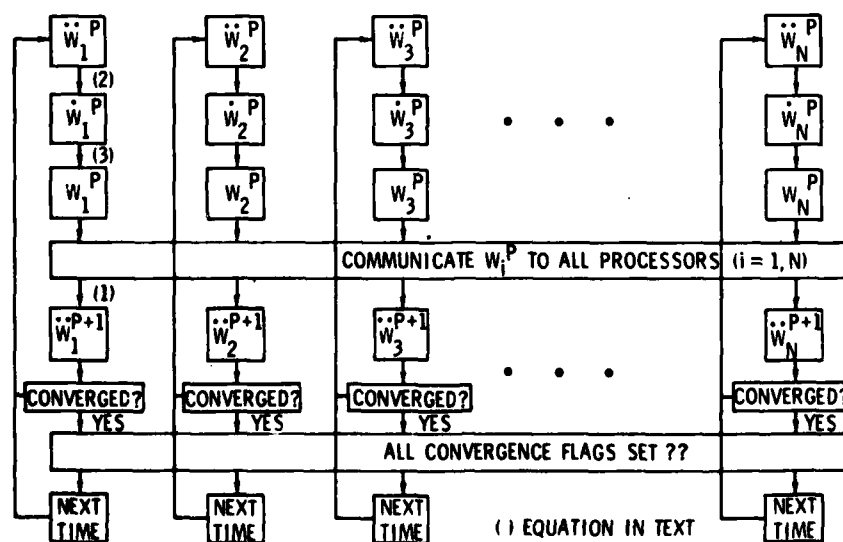


Figure 23. Parallel Integration Method for Transient Response (processors across, time down)

### BEAM GRILLAGE RESPONSE

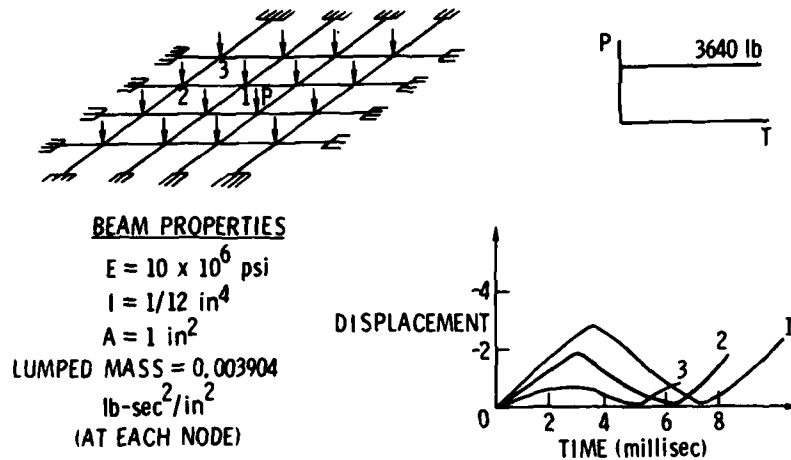


Figure 24. Beam Grillage Response

### COMPUTATIONAL SPEEDUP: BEAM GRILLAGE

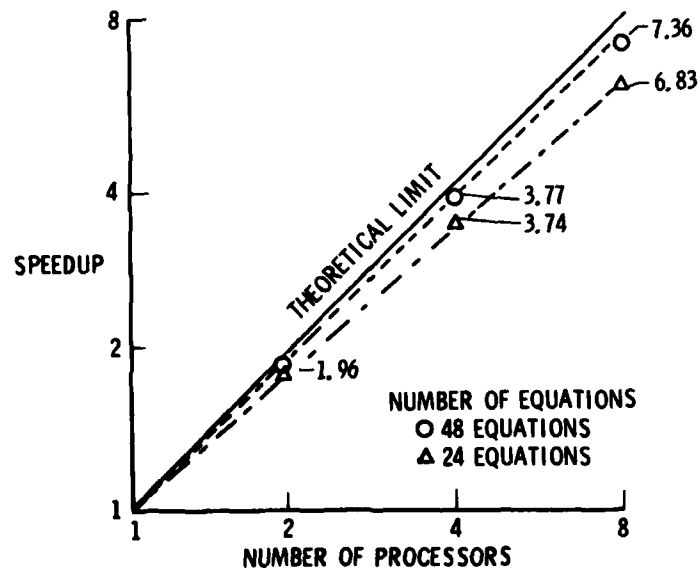


Figure 25. Computation Speedup vs. Number of Processors for Beam Grillage

### EXTRAPOLATION TO FUTURE ANALYSIS REQUIREMENTS

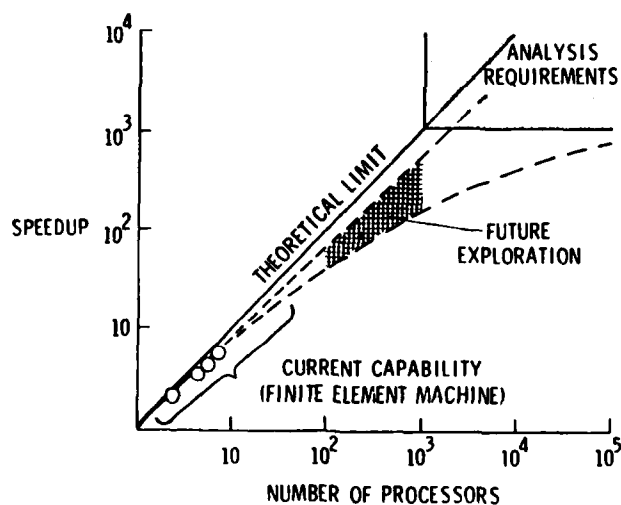


Figure 26. Computation Speedups Possible through Use of Concurrent Computers

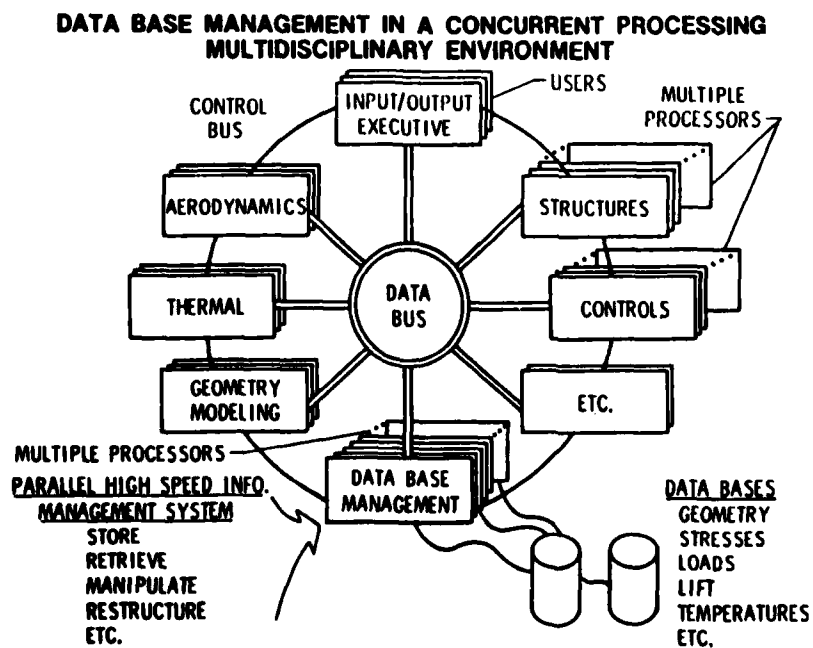


Figure 27. Data Base Management in a Concurrent Processing Multidisciplinary Environment

### **NEEDED FINITE ELEMENT RESEARCH**

- ENGR
  - REQUIREMENTS FOR FUTURE F. E. SYSTEM
  - MODELING STRATEGIES
  - IMPROVED ANALYTICAL SIMULATION
- CSC
  - CONCURRENT PROCESSING STRATEGIES
  - DISTRIBUTED PROCESSING AND CONTROL
  - HIGH SPEED COMMUNICATIONS AND N/W
  - DATA MANAGEMENT
  - 3D GEOMETRY MODELING/DISPLAY
  - F. E. SYSTEM DESIGN
  - SPECIALIZED H/W, S/W DESIGNS FOR F. E. APPLICATIONS
- MGT
  - MATCHING H/W, S/W, ALGORITHMS ALTERNATIVES FOR EFFECTIVE PROBLEM SOLUTION
  - STANDARDS FOR INTEGRATING DISTRIBUTED F. E. SYSTEMS
- NUMERICAL ALGORITHMS
  - IDENTIFICATION OF AND METHODS FOR PARALLELISM
  - DISTRIBUTED F. E. COMPUTATIONS
  - DECOMPOSITION OF CALCULATIONS ONTO ARRAYS OF DIFFERENT PROCESSING
  - DEVELOPMENT OF FIRMWARE ALGORITHMS

Figure 28. Research Needed to Support Future Structural Analysis Capabilities

REPORT DOCUMENTATION PAGE			
<b>1. Recipient's Reference</b>	<b>2. Originator's Reference</b> AGARD-R-726	<b>3. Further Reference</b> ISBN 92-835-1522-6	<b>4. Security Classification of Document</b> UNCLASSIFIED
<b>5. Originator</b>	Advisory Group for Aerospace Research and Development North Atlantic Treaty Organization 7 rue Ancelle, 92200 Neuilly sur Seine, France		
<b>6. Title</b>	THE INFLUENCE OF LARGE SCALE COMPUTING ON AIRCRAFT STRUCTURAL DESIGN		
<b>7. Presented at</b>			
<b>8. Author(s)/Editor(s)</b> Various			<b>9. Date</b> April 1986
<b>10. Author's/Editor's Address</b> Various			<b>11. Pages</b> 70
<b>12. Distribution Statement</b>	This document is distributed in accordance with AGARD policies and regulations, which are outlined on the Outside Back Covers of all AGARD publications.		
<b>13. Keywords/Descriptors</b>  <div style="display: flex; justify-content: space-between;"> <div> Airframes Design Data processing </div> <div> Computation Computers </div> </div>			
<b>14. Abstract</b>  <p>An outcome of the AGARD Working Group on Large Scale Computing was the institution by the Structures and Materials Panel of a Sub-Committee to consider this topic in relation to Aircraft Structural Design. This publication contains papers on various aspects of the topic heard by the Sub-Committee plus a summary. These cover likely developments in both soft- and hardware, and their applications to the Design process.</p> <p>This Report was sponsored by the Structures and Materials Panel of AGARD.</p>			



**AGARD**

**NATO OTAN**

**7 RUE ANCELLE · 92200 NEUILLY-SUR-SEINE  
FRANCE**

**Telephone (1) 47.45.08.10 · Telex 810178**

**DISTRIBUTION OF UNCLASSIFIED  
AGARD PUBLICATIONS**

AGARD does NOT hold stocks of AGARD publications at the above address for general distribution. Initial distribution of AGARD publications is made to AGARD Member Nations through the following National Distribution Centres. Further copies are sometimes available from these Centres, but if not may be purchased in Microfiche or Photocopy form from the Purchase Agencies listed below.

**NATIONAL DISTRIBUTION CENTRES**

**BELGIUM**

Coordonnateur AGARD - VSL  
Etat-Major de la Force Aérienne  
Quartier Reine Elisabeth  
Rue d'Evere, 1140 Bruxelles

**CANADA**

Defence Scientific Information Services  
Dept of National Defence  
Ottawa, Ontario K1A 0K2

**DENMARK**

Danish Defence Research Board  
Ved Idraetsparken 4  
2100 Copenhagen Ø

**FRANCE**

O.N.E.R.A. (Direction)  
29 Avenue de la Division Leclerc  
92320 Châtillon

**GERMANY**

Fachinformationszentrum Energie,  
Physik, Mathematik GmbH  
Kernforschungszentrum  
D-7514 Eggenstein-Leopoldshafen

**GREECE**

Hellenic Air Force General Staff  
Research and Development Directorate  
Holargos, Athens

**ICELAND**

Director of Aviation  
c/o Flugrad  
Reykjavik

**ITALY**

Aeronautica Militare  
Ufficio del Delegato Nazionale all'AGARD  
3 Piazze Adenauer  
00144 Roma/EUR

**LUXEMBOURG**

See Belgium

**NETHERLANDS**

Netherlands Delegation to AGARD  
National Aerospace Laboratory, NLR  
P.O. Box 126  
2600 AC Delft

**NORWAY**

Norwegian Defence Research Establishment  
Attn: Biblioteket  
P.O. Box 25  
N-2007 Kjeller

**PORTUGAL**

Portuguese National Coordinator to AGARD  
Gabinete de Estudos e Programas  
CLAPA  
Base de Alfragide  
Alfragide  
2700 Amadora

**TURKEY**

Department of Research and Development (ARGE)  
Ministry of National Defence, Ankara

**UNITED KINGDOM**

Defence Research Information Centre  
Kantigern House  
65 Brown Street  
Glasgow G2 8EX

**UNITED STATES**

National Aeronautics and Space Administration (NASA)  
Langley Research Center  
M/S 180  
Hampton, Virginia 23665

**THE UNITED STATES NATIONAL DISTRIBUTION CENTRE (NASA) DOES NOT HOLD  
STOCKS OF AGARD PUBLICATIONS, AND APPLICATIONS FOR COPIES SHOULD BE MADE  
DIRECT TO THE NATIONAL TECHNICAL INFORMATION SERVICE (NTIS) AT THE ADDRESS BELOW.**

**PURCHASE AGENCIES**

**Microfiche or Photocopy**

National Technical  
Information Service (NTIS)  
5285 Port Royal Road  
Springfield  
Virginia 22161, USA

**Microfiche**

ESA/Information Retrieval Service  
European Space Agency  
10, rue Mario Nille  
75015 Paris, France

**Microfiche or Photocopy**

British Library Lending  
Division  
Boston Spa, Wetherby  
West Yorkshire LS23 7BQ  
England

Requests for microfiche or photocopies of AGARD documents should include the AGARD serial number, title, author or editor, and publication date. Requests to NTIS should include the NASA accession report number. For bibliographical references and abstracts of AGARD publications see below in the following format:

**Scientific and Technical Aerospace Reports (STAR)**

Published by NASA Langley and Dryden

Information Report

AGARD-AR-1234

Washington, D.C., USA

**Government Reports Administration (GRA)**

Published by the National Technical

Information Service, Springfield

Virginia 22161, USA

Printed by Specialist Printing Services Limited  
40 Chigwell Lane, Loughborough, Essex IG10 3TZ

ISBN 92-835-1522-6

DATE  
FILMED  
0-8